



**GLOBAL EN ROUTE BASING
INFRASTRUCTURE LOCATION MODEL**

GRADUATE RESEARCH PROJECT

Alexander Miravite, Jr.
Major, USAF

Charles F. Schlegel
Major, USAF

AFIT/IOA/ENS/06-08

**DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY**

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

The views expressed in this paper are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

**GLOBAL EN ROUTE BASING
INFRASTRUCTURE LOCATION MODEL**

GRADUATE RESEARCH PROJECT

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Operations Analysis

Alexander Miravite, Jr.
Major, USAF

Charles F. Schlegel
Major, USAF

May 2006

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

**GLOBAL EN ROUTE BASING
INFRASTRUCTURE LOCATION MODEL**

Alexander Miravite, Jr.
Major, USAF

Charles F. Schlegel
Major, USAF

Approved:

Lt Col Raymond W. Staats (Chairman)

Date

Abstract

The backbone of the air transportation system for the US military is the En Route System (ERS), a collection of personnel and equipment at bases throughout the world. The location of these bases in the ERS bears the marks of the Cold War legacy of the latter half of the twentieth century. While many changes to the ERS have taken place in the last 20 years, it still has a predominantly East-West orientation in the Northern Hemisphere with little capability in many parts of the world.

This research provides the decision makers (DMs) with a clear methodology for evaluating the best locations for new en route bases in the ERS. It looks at the capability of each en route airfield alternative using 27 different measures. It then examines the National Security Import of the destination regions an en route airfield could service, along with the probability of events in those regions requiring military action, including humanitarian and other missions. This is accomplished using Value Focused Thinking (VFT) and decision analysis tools, with the goal of providing insight to the DMs to make informed decisions. This methodology is then packaged in a powerful Excel-based, user interface, allowing quick analysis of multiple scenarios. The Global En Route Basing Infrastructure Location model (GERBIL) is currently being used by TRANSCOM/J5 to prioritize Cooperative Security Locations around the globe for possible TRANSCOM mobility en route locations.

We dedicate this Graduate Research Project to our families. Without the strong support, encouragement, and love of our wives, we could have not accomplished a fraction of what we have been able to achieve.

Acknowledgments

We would like to thank Capt Katie Tharaldson for her hard work in developing the value hierarchy which serves as the foundation for the Tactical Submodel in our research

We would like to thank Lt Col Weir for the computer code template used in developing the single dimension value functions in the GERBIL model

We would like to thank Dr. James Moore and the funding we received from the Air Force Office of Scientific Research, grant QAF 185025203206.

We would like to thank Lt Col Diane Allen and TRANSCOM/J5 for sponsoring this line of mobility research.

Above all, we would like to thank our advisor, Lt Col Staats, for his guidance and mentorship in this process.

Alex ‘R2’ Miravite and Charles ‘Doc’ Schlegel

Table of Contents

Abstract.....	iv
List of Figures.....	x
List of Tables	xii
I. Draft Article.....	1
Background	1
En Route System.....	1
Value Focused Thinking.....	3
Tactical Sub-Model.....	4
GERBIL Operational Value Hierarchy.....	11
Probability of Alternative Utilization	12
National Security Import of Destination Region	16
Aggregation	18
GERBIL	19
Set Up	19
Case Study	23
Set Up	23
Conclusions and Future Research.....	28
Acknowledgements.....	29
Author Statement	29
II. Extended Narrative.....	30
Changes to the Tactical Sub-model	30
Background.....	30
Critical Leg Measure	30
Maximum On Ground (MOG).....	31
Fuel Resupply	32
Uncertainty	33
III. Future Research	35
Throughput.....	35
Multiple Origins.....	35
Risk Tolerance	36
Three Leg Flights, Two En Route Stops.....	38
Appendix 1. Operational Model Data	40
Origin/Destination Airfields Data.....	40
Enroute Alternatives Data.....	41
Tactical Submodel Value Functions	44
Tactical Submodel Weighting.....	49
Operational Model Value Functions	50
Operational Model Weighting	51
National Security Measure Definitions.....	51

Appendix 2. GERBIL User's Guide	52
About GERBIL	58
Model Development.....	58
Tactical Submodel	58
Operational Value Hierarchy	59
Software Notes.....	59
Enabling Macros	59
Enabling ActiveX Controls.....	60
Closing Error	60
Application Interface	60
Saving GERBIL Output.....	63
Saving/Restoring Model	63
Integrated User Help.....	64
Origin/Destination Airfields.....	64
Latitude/Longitude format	64
Destination Attributes	65
Probability of Utilization	65
National Security	65
Adding Uncertainty.....	66
Throughput	66
En Route Alternatives Module	66
Measure Definitions.....	67
Airfield Input Form.....	67
Latitude/Longitude Format	67
Continuous Measures.....	68
Categorical Measures.....	68
Assigning Probability	68
Distance calculations	69
Critical Leg	69
Delta Flight Length.....	69
Within the Lens Determination.....	70
View Tactical Submodel Module.....	71
Measure Definitions.....	71
Value Functions	71
Continuous Increasing	72
Continuous Decreasing	73
Categorical	73
Save/Restore Value Functions	74
Weighting.....	75
Local vs Global	75
Weighting by Branch	75
Swing Weighting	75
Showing All Weights.....	76
Global Weights Chart	77
Sorted Global Weights Chart.....	78
Save/Restore Weights	78

Running the Tactical Submodel	79
Value Scores	79
Rankings Chart.....	80
Displaying by different Tiers	81
Copying Value/Rank Chart Results to File.....	81
Sensitivity Analysis	81
Local Proportional	81
Global Proportional.....	82
Copying Sensitivity Analysis Results to File	82
Running an Operational Scenario.....	84
Origin Selection	85
Operational Value Hierarchy	85
Weighting the Operational Value Hierarchy.....	85
Swing Weighting	85
Value Functions	85
Probability of Utilization and the Prospect Function.....	85
National Security Index Function	86
Operational Scenario Results.....	86
Scenario Scores.....	86
Scenario Top 10 Chart	87
Copying Scores/Chart Results to File	87
Scenario Sensitivity Analysis	88
Copying Sensitivity Analysis Results to File.....	88
Appendix 3. GERBIL Visual Basic Code	90
Charts/Worksheets/Workbook Code	91
User Forms Code	93
Modules Code	145
Bibliography	198

List of Figures

Figure 1: Graphical display of East-West bias of current ERS	2
Figure 2: Current European en route bases and the ‘lens’ from the Enroute Strategic Plan (McVicker, 2002).....	4
Figure 3: Tactical sub-model from GERBIL	6
Figure 4: SDVF for CRITICAL LEG measure (nautical miles).....	8
Figure 5: SDVF for FUEL RESUPPLY measure (millions of gallons per day)	9
Figure 6: Operational value hierarchy with branch weightings.....	12
Figure 7: Prospect Function for Probability of Utilization of Destination	13
Figure 8: SDVF for National Security Import	18
Figure 9: GERBIL data input screen for en route alternatives	21
Figure 10: Global weights for tactical sub-model.....	22
Figure 11: GERBIL results of tactical sub-model run, Ramstein AB Germany to Hoseo Kutako Intl Namibia	24
Figure 12: GERBIL results of operational scenario run with Ramstein AB Germany as APOE	26
Figure 13: GERBIL sensitivity analysis	27
Figure 14: MOG SDVF	44
Figure 15: Critical Leg SDVF.....	44
Figure 16: Delta Flight Length SDVF	44
Figure 17: Alternate Airfields SDVF.....	44
Figure 18: Fuel Storage SDVF	44
Figure 19: Fuel Resupply SDVF	44
Figure 20: Diplomatic Clearance SDVF.....	45
Figure 21: Force Protection SDVF	45
Figure 22: Department of State SDVF.....	45
Figure 23: Military Cooperation SDVF.....	45
Figure 24: Seaport SDVF	45
Figure 25: Railroad SDVF	45
Figure 26: Road System SDVF	46
Figure 27: Commercial Airport SDVF	46
Figure 28: Lodging SDVF	46
Figure 29: Dining SDVF.....	46
Figure 30: Medical SDVF.....	46
Figure 31: Communications SDVF	46
Figure 32: Power SDVF.....	47
Figure 33: Potable Water SDVF	47
Figure 34: Sewer SDVF.....	47
Figure 35: Mountainous SDVF.....	47
Figure 36: Altitude SDVF.....	47
Figure 37: Weather SDVF	47
Figure 38: Temperature SDVF	48
Figure 39: Urban Areas SDVF	48
Figure 40: Terrain SDVF.....	48
Figure 41: Tactical Submodel Hierarchy	49
Figure 42: Tactical Submodel Weighting	49

Figure 43: Probability of Utilization SDVF.....	50
Figure 44: National Security SDVF.....	50
Figure 45: Operational Model Weighting.....	51
Figure 46: Tactical Submodel Diagram	58
Figure 47: Operational Value Hierarchy Diagram.....	59
Figure 48: Enabling Macros.....	59
Figure 49: Enabling ActiveX Controls Dialog	60
Figure 50: Closing error message	60
Figure 51: GERBIL Main Menu.....	61
Figure 52: Save/Restore Model Dialog.....	63
Figure 53: Origin/Destination Airfield Spreadsheet	64
Figure 54: Add Destination Airfield Dialog	65
Figure 55: En route Alternatives Module Spreadsheet	66
Figure 56: Airfield Input Form Dialog	67
Figure 57: Triangle Distribution Dialog	68
Figure 58: Empirical Distribution Dialog	69
Figure 59: View Tactical Submodel Module.....	71
Figure 60: Continuous Increasing SDVF.....	72
Figure 61: Continuous Decreasing SDVF	73
Figure 62: Categorical SDVF	74
Figure 63: Save/Restore Value Function Dialog	74
Figure 64: Sample Local Weighting by Branch	75
Figure 65: Show All Weights Spreadsheet	77
Figure 66: Global Weights Chart.....	78
Figure 67: Sorted Global Weights Chart	78
Figure 68: Save/Restore Weights.....	78
Figure 69: Run Tactical Submodel Menu.....	79
Figure 70: Value Scores Spreadsheet	80
Figure 71: Tactical Score Rankings Chart	80
Figure 72: Sample Local Proportional Sensitivity Analysis.....	82
Figure 73: Sample Global Proportional Sensitivity Analysis	82
Figure 74: Operational Scenario Menu.....	84
Figure 75: Prospect Function for Probability of Utilization	86
Figure 76: Scenario Score Matrix	87
Figure 77: Scenario Rankings Chart.....	87
Figure 78: Sample Scenario Sensitivity Analysis Chart.....	88

List of Tables

Table 1: Tactical Sub-model Measures, Corresponding Value Functions, and Original Global Weightings	6
Table 2: Constructed, Proxy Scale for National Security Import	16
Table 3: Origin Airfield Data.....	20
Table 4: Destination Airfield Data.....	20
Table 5: GERBIL results of operational scenario run with Ramstein AB Germany as the APOE	26
Table 6: Revised Fuel Weighting	33
Table 7: Multiattribute Risk Tolerance, $\rho_m = 0.2$	37
Table 8: Multiattribute Risk Tolerance, $\rho_m = 1$	37
Table 9: Multiattribute Risk Tolerance, $\rho_m = 10$	38
Table 10: Multiattribute Risk Tolerance, $\rho_m = 10000$	38
Table 11: Origin Airfield Data.....	40
Table 12: Destination Airfield Data.....	40
Table 13: Enroute Alternatives Data Table A.....	41
Table 14: Enroute Alternatives Data Table B.....	41
Table 15: En Route Alternatives Data Table C	42
Table 16: En Route Alternatives Data Table D	42
Table 17: En Route Alternatives Data Table E.....	43
Table 18: National Security Measure Definitions	51

GLOBAL EN ROUTE BASING INFRASTRUCTURE LOCATION MODEL

I. Draft Article

Background

En Route System

“To deter threats against or assist in the defense or pursuit of US interests, the United States maintains forces, organizations, and processes, necessary to conduct and sustain air mobility operations globally, rapidly, and on a scale not matched by any other nation”. This overarching Joint Doctrine from Joint Publication 3-17 (2005) is the foundation for one of the United States Air Force’s (USAF) six distinctive capabilities, Rapid Global Mobility. According to the USAF (2006), “being able to respond quickly and decisively anywhere we’re needed is key to maintaining rapid global mobility”. The success of Rapid Global Mobility relies on the health of the Strategic Airlift En Route System (ERS). This system, and its capabilities, is not static but rather is always changing.

For instance, over the past 15 years, Air Mobility Command (AMC) has reduced the number of bases in the ERS from 39 locations to 12 permanent locations today (McVicker, 2002). As of 2006, these main locations include six in Europe (Spangdahlem Air Base (AB) and Ramstein AB Germany, Moron AB and Rota AB Spain, Fairford AB and Mildenhall AB United Kingdom) and six in the Pacific (Elmendorf Air Force Base (AFB) Alaska, Hickam AFB Hawaii, Andersen AFB Guam, Kadena AB and Yokota AB Japan, and Osan AB Korea). It should be noted that there are other locations with a small AMC presence, but these smaller bases are not considered main en route bases because of their capability limitations.

While the attacks of 9/11 have significantly changed the national security environment, the ERS still bears the clear marks of the Cold War landscape. Our capability to rapidly deploy

assets around the globe is largely reliant on bases established after World War II in Western Europe and the Pacific. These bases largely support the deployment, sustainment, and redeployment of personnel and equipment along an East – West route. This East – West bias in the current ERS is evident in Figure 1, which displays the current global reach of the ERS (McVicker, 2002). The current ERS structure is configured to account for the 3,500 nm range limitation of a C-17A (without extended range fuel tanks), carrying a payload of 90,000 pounds (McVicker, 2002).

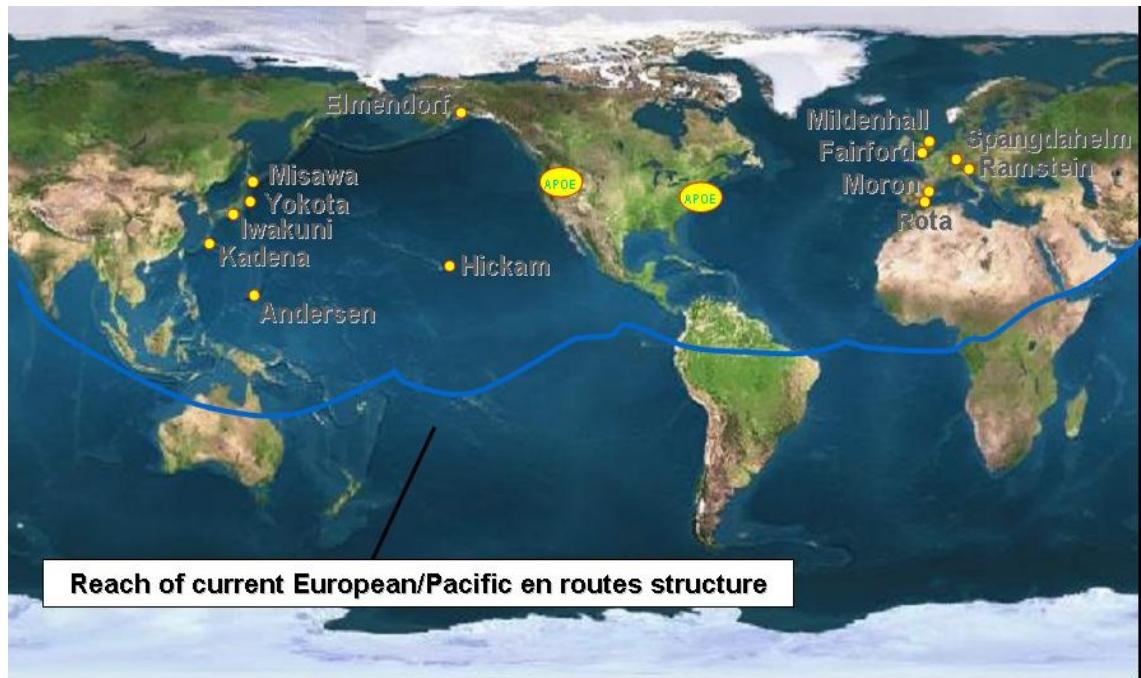


Figure 1: Graphical display of East-West bias of current ERS

The dynamics of the Global War on Terror (GWOT) and the ever-evolving geopolitical climate require us to develop a methodology to ensure the ERS remains able to support the key capability of Rapid Global Mobility outside these old Cold War boundaries. In a perfect world, Transportation Command (TRANSCOM, the joint command responsible for all DoD transportation) would build new en route bases around the world as needed to fill in the current gaps and provide the DoD with global coverage. Fiscal reality alone, however, makes this an

impossible option. The challenge is to identify possible en route bases and rank them according to the value they would bring to the ERS. This article examines a way to help achieve this goal by providing a robust methodology for senior leaders to choose new location(s) to add to the current ERS.

Value Focused Thinking

For this complex decision problem, a Value Focused Thinking approach provides the systematic process that the Decision Maker (DM) requires to make an informed decision. The power of VFT is that it centers the decision process on the evaluation of objective criteria, and value tradeoffs, rather than on a set of predetermined alternatives. Keeney (1992) describes the latter process as alternative-focused thinking and describes its limitations to great extent. The major flaw according to Keeney is “the fundamental objectives and the decision context are not usually made explicit with alternative-focused thinking” (Keeney, 1992).

The current methodology to find new bases for the ERS has been to focus on what is referred to as “the lens.” The lens refers to the overlapping of two circles, one extending out 3,500 nm from the origin in the United States and the other extending out 3,500 nm from the final destination. Due to the geopolitical considerations prior to 9/11, the two destinations given the most emphasis were North Korea and Iraq. While the lens shows the best area for a specific origin and destination pair, it limits the observer to looking at the utility of an alternative in only one circumstance as shown in Figure 2 (McVicker, 2002). Also, because of the emphasis put on North Korea and Iraq, it unintentionally drew focus away from other areas, as illustrated in Figure 1, which are now critical to the success of the GWOT.



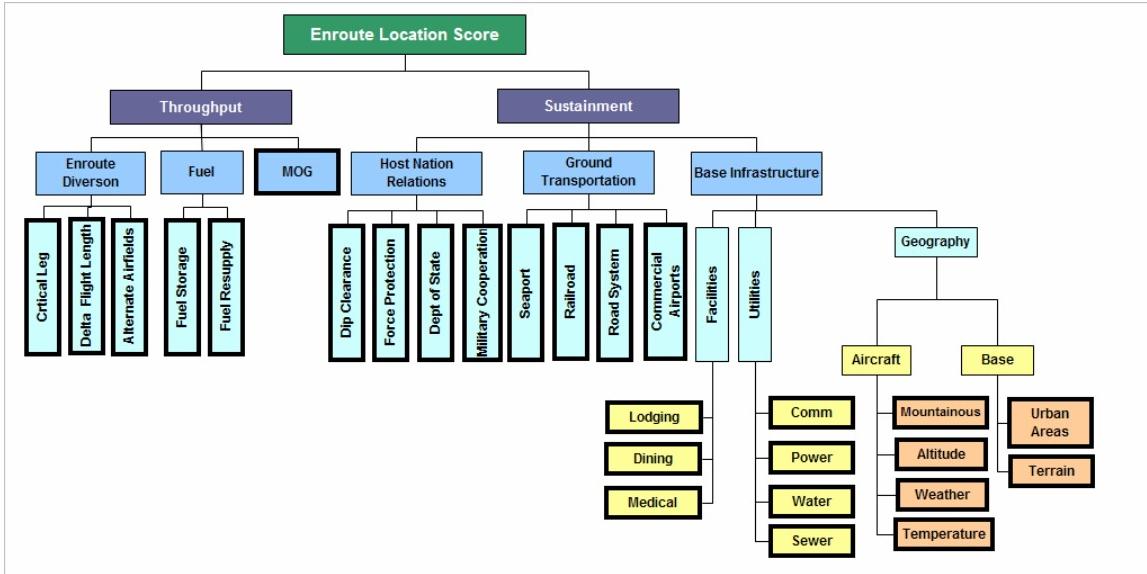
Figure 2: Current European en route bases and the ‘lens’ from the Enroute Strategic Plan (McVicker, 2002)

The challenge for the ERS problem is to develop a methodology which can evaluate a possible en route site for its overall value to the National Security Strategy (NSS) and National Military Strategy (NMS). This is the fundamental objective of this research, to maximize the overall value to the NSS and NMS by strengthening the ERS’s capability to provide Rapid Global Mobility. Prior to evaluating any possible alternatives, however, a value hierarchy was developed to ascertain what DMs value in an en route airfield. This research builds upon the value hierarchies proposed by Sere (2005) and Tharaldson (2006). The value hierarchy developed in the latter was modified to be more comprehensive, and this revised value hierarchy is referred to hereafter as the tactical sub-model. This tactical sub-model was then embedded into a higher-level model, called the operational value hierarchy, which is discussed in further detail in this paper.

Tactical Sub-Model

The tactical sub-model, shown in Figure 3, was adapted from the value hierarchy initially developed by Tharaldson (2006). It measures the physical attributes which are valued in an en route base. Given a specific origin and specific destination, the tactical sub-model calculates a value for each of the twenty-seven different measures of an alternative. These twenty-seven measures, each with a single dimensional value function (SDVF) associated with it, are contained in a five-tier hierarchy. The measures, their type of value function, and their corresponding, original global weights are shown in Table 1.

The first tier of the tactical sub-model is broken into two objectives, Throughput and Sustainment. Throughput measures the capability of an en route airfield to enable the flow of cargo and personnel through the airfield and on to the final destination. This capability to keep aircraft, and their associated cargo and passengers, moving quickly through the system is essential for DoD operations. Throughput is further broken down in the second tier as three sub-objectives: En Route Diversion, Fuel, and MOG. Each of these three sub-objectives, and the measures for them, provide insight on the effectiveness of an alternative for the throughput of cargo and personnel. Sustainment measures the capability of the alternative to continue to support operations on a permanent basis. Prior to sinking significant resources into a new en route airfield for the ERS, it is important to know that the airfield is suited for long term use. Sustainment is further broken down into three sub-objectives: Host Nation Relations, Ground Transportation, and Base Infrastructure. Each of these three sub-objectives, and the measures for them, provide insight on the viability of an alternative for long term use.



**Figure 3: Tactical sub-model from GERBIL.
The twenty-seven measures have bold borders**

Table 1: Tactical Sub-model Measures, Corresponding Value Functions, and Original Global Weightings

Measure	Type of Value Function	Original Global Weight
Critical Leg	Continuously decreasing	0.048
Delta Flight Length	Continuously decreasing	0.071
Alternate Airfields	Continuously increasing	0.024
Fuel Storage	Continuously increasing	0.009
Fuel Resupply	Continuously increasing	0.027
MOG	Continuously increasing	0.072
Diplomatic Clearance	Categorical	0.037
Force Protection	Categorical	0.111
Dept of State	Categorical	0.074
Military Cooperation	Categorical	0.111
Seaport	Continuously decreasing	0.009
Railroad	Continuously decreasing	0.009
Road System	Continuously decreasing	0.022
Commercial Airport	Continuously decreasing	0.043
Lodging	Categorical	0.032
Dining	Categorical	0.016
Medical	Binary	0.003
Communications	Categorical	0.005
Power	Categorical	0.031
Potable Water	Categorical	0.021
Sewer	Categorical	0.021
Mountainous	Binary	0.034
Altitude	Binary	0.017
Weather	Categorical	0.068
Temperature	Binary	0.017
Urban Areas	Binary	0.057
Terrain	Binary	0.011

Changes to the original hierarchy included the major modification of one SDVF, the deletion of a measure, the addition of a measure, and the introduction of risk assessment for a number of the measures. Detailed development of the hierarchy used in the paper can be found in Tharaldson (2006). Detailed SDVF and weight factors for the modified model are contained in Miravite and Schlegel (2006). The SDVF for the critical leg measure was changed from continuous increasing to continuous decreasing, with the revised SDVF shown in Figure 4. Since the critical leg is the greater of the distance from the origin to the alternative or the alternative to the destination, a smaller number is more desirable. The lower bound for his measure is given as 1,751 nm. If this distance is less than the total distance from the origin to destination would be less than the 3,500 nm range and within the unrefueled range of the aircraft, thus an en route airfield would not be required.

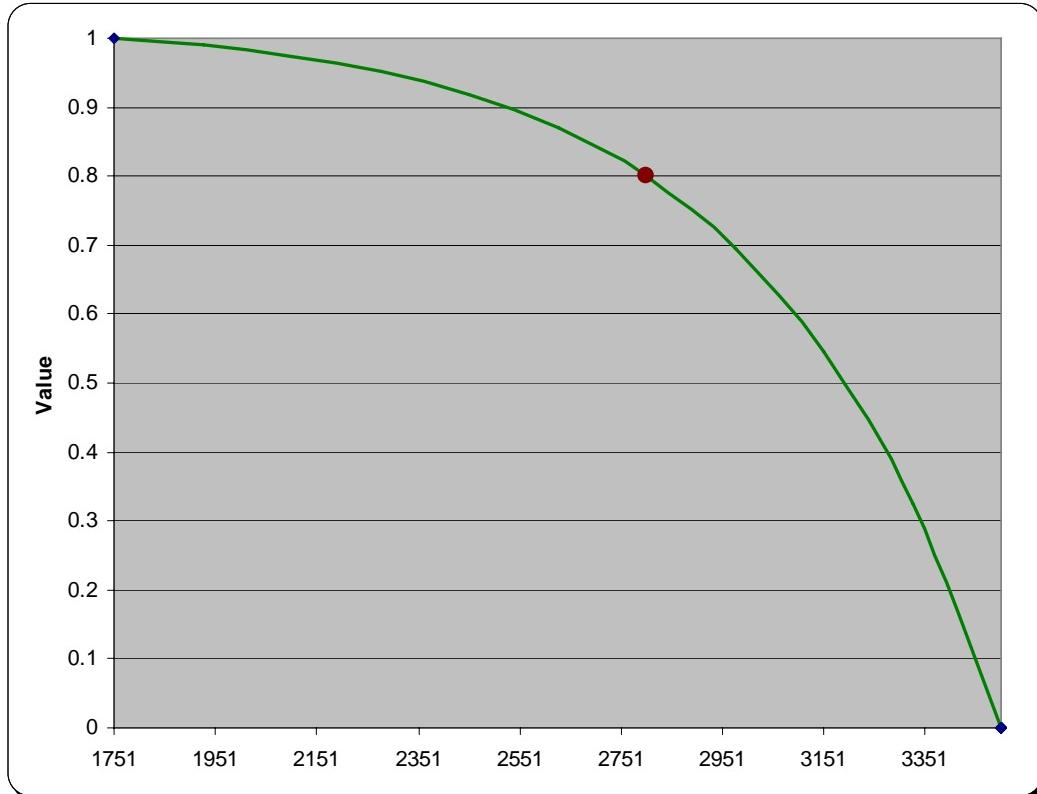


Figure 4: SDVF for CRITICAL LEG measure (nautical miles)

The measure deleted was Working MOG. According to the En Route Strategic Plan (McVicker, 2002), “the Working MOG is generally accepted to be one half the Parking MOG”. Since this is the commonly accepted definition of Working MOG, having a separate measure for both the Working and Parking MOG violated the principle of mutual exclusiveness required in the model. Therefore, the measure Working MOG was deleted and the global weighting of the Parking MOG was adjusted accordingly.

Tharaldson’s original model included only one measure for fuel--Fuel Storage. The Fuel Storage measure is essential to provide a steady supply of fuel at the onset of operations and to allow operations to continue when fuel resupply is affected by weather, maintenance, or other factors such as local holidays. In addition, the fuel storage provides a buffer for surge operations above the daily fuel resupply rate at an en route airfield. While the Fuel Storage measure remains a very important factor in the model, it was obvious a second measure for the fuel resupply rate was needed to account for how quickly fuel could be replenished. The new Fuel Resupply measure, shown in Figure 5, is a continuously increasing SDVF and takes into account the large amount of fuel which is required to sustain operations over a long period of time.

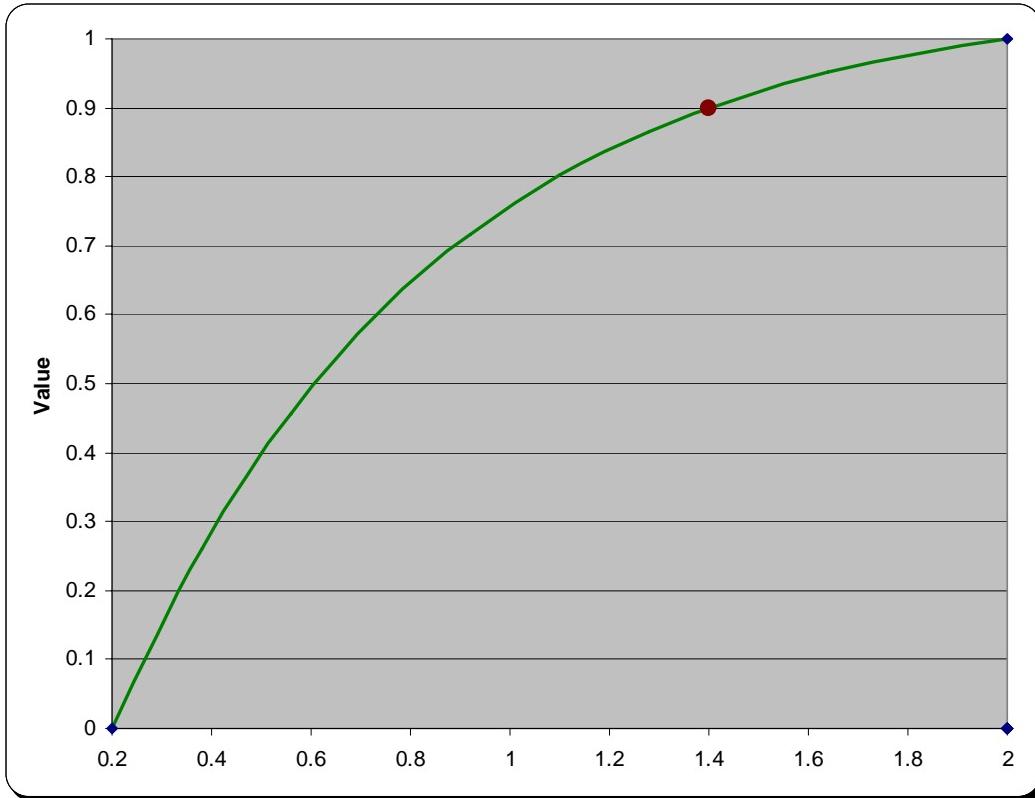


Figure 5: SDVF for FUEL RESUPPLY measure (millions of gallons per day)

A major addition to the tactical sub-model was the introduction of uncertainty to previously deterministic measures. The decision was made, however, to not include multiattribute risk tolerance into the model. Kirkwood (1997) discusses that in 90 to 95% of the decisions, the preferred alternative is the same when using expected value and expected utility. He goes on to state that if a sensitivity analysis for ρ_m from 0.2 to infinity does change the preferred alternative, which was true in this model's case study, then risk aversion generally does not have to be considered. For further discussion on the decision not to include multiattribute risk tolerance in this model see Miravite and Schlegel (2006).

For uncertainty in the model, the two probability distributions utilized were triangular and empirical. For example, if the Fuel Storage measure is determined to not be an exact known figure, the user inputs a lower bound amount, the most likely amount, and an upper bound

amount. A triangular distribution is generated using these parameters, and the expected value is input to the model. The triangular function was chosen for its simplicity and ease of understanding. While the distribution is simple, it is also appropriate since the type of data and the knowledge of the data lends itself to the triangular distribution. This triangular, stochastic function is available for all but two of the continuous measures in the tactical sub-model.

Continuous measures Critical Leg and Delta Flight Length are deterministically computed using great circle route distance calculations. A number of the measures in the tactical submodel, however, are not continuous but rather categorical. If the categorical value for a measure is not known with certainty, the user can enter a discrete empirical distribution specifying the probabilities that the measure takes on the corresponding categorical values. When a function is categorical with a predetermined number of outcomes, the empirical distribution is the only stochastic distribution which makes sense.

With the tactical sub-model now in place, the value for each of these twenty-seven measures, $v_i^T(a_i^T)$, for an alternative, A , is weighted by w_i^T and summed to calculate the value of the alternative, $V_T(A_d)$, given a specific origin and destination. The following definitions are applicable to the mathematical formulation of the tactical sub-model:

D	set of all possible destinations
A	an en route alternative
D_A	$D_A \subset D$. The set of destinations feasible to A and whose great circle distance between the origin and destination is greater than the maximum critical leg value (i.e. an en route airfield is required).
d	specific destination
a_i^T	measurement level of attribute i of the tactical sub-model
$v_i^T(a_i^T)$	SDVF of attribute i of the tactical sub-model
w_i^T	global weighting factor for attribute i of the tactical sub-model

The tactical sub-model score, $V^T(A_d)$, is calculated using equation (1) and then used as an input to the higher-level, larger operational value hierarchy.

$$V^T(A_d) = \begin{cases} \sum_i w_i^T v_i^T(a_{id}^T) & \text{where } \sum_i w_i^T = 1, \text{ if } d \in D_A \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

GERBIL Operational Value Hierarchy

The tactical sub-model enables the identification of airfields from around the world and to evaluate them given a specific origin and destination. A limitation of this model is that it was not designed to evaluate operational scenarios. The operational value hierarchy incorporates the tactical sub-model as an objective, adding two more considerations; the probability of a destination being utilized for operations in the region and the National Security value of the region which would be accessed from the destination. The research also took what was a deterministic tactical sub-model and introduces uncertainty into both the tactical, as discussed previously, and operational value hierarchies. Finally, this research develops a customizable user-friendly computer interface to automate the evaluations, given a specific origin.

The operational value hierarchy has three branches and is shown along with the assessed branch weights in Figure 6. The first branch is the tactical sub-model, discussed in the foregoing section, and represents the tactical value score of an alternative for a specific origin and destination. The second branch is the probability of an event occurring in the destination region which would require the use of the destination airfield, and thus the en route alternative airfield, to be utilized. The third branch measures the value of the destination region in terms of National Security. For each alternative, given a specific origin and destination, value scores are derived for each of the three branches of the operational value hierarchy: the tactical sub-model, $V_T(A_d)$,

the probability of utilizing the destination airfield, $V^P(A_d)$, and the National Security importance ascribed to the destination region $V^N(A_d)$.

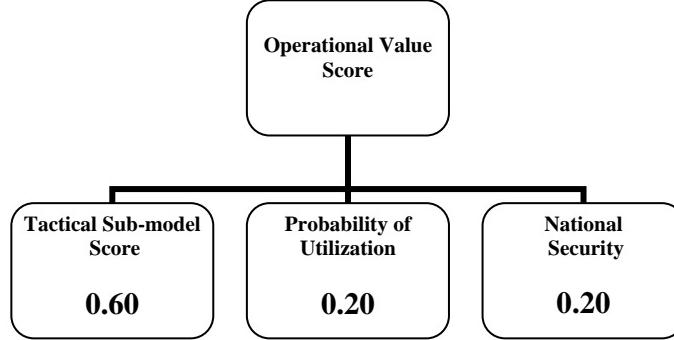


Figure 6: Operational value hierarchy with branch weightings

Probability of Alternative Utilization

The value function of the second branch, the probability of an event occurring in the destination region which would require the alternative to be utilized, uses Prospect Theory from the research of Tversky and Kahneman (1992). The SDVF for the probability of utilizing the destination airfield, P_d , is given a value, $V^P(A_d)$, by using this Prospect Function, $\pi(P)$, shown in equation 2.

P_d Subjective probability elicited from DM

$\pi(P_d)$ Value of the subjective probability from the Prospect Function

$$\pi(P_d) = 1.89799P_d - 3.55955P_d^2 + 2.662549P_d^3 \quad (2)$$

$$V^P(A_d) = \begin{cases} \pi(P_d) & d \in D_A, \\ 0 & otherwise \end{cases} \quad (3)$$

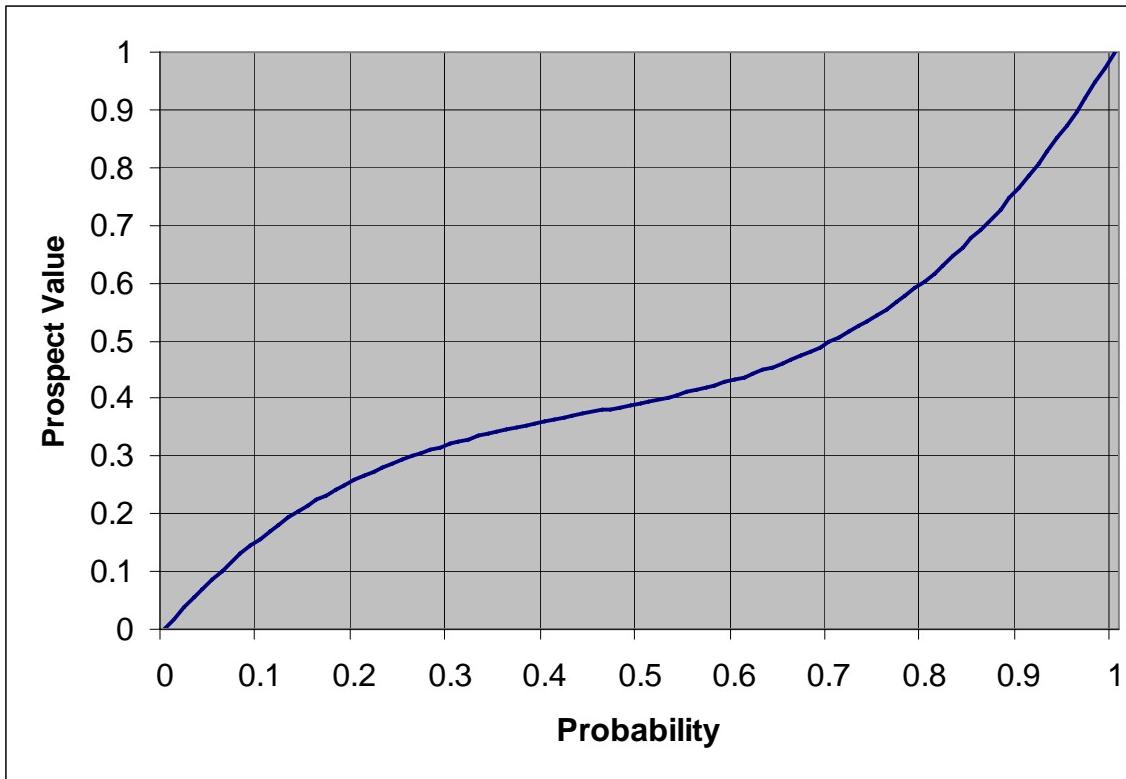


Figure 7: Prospect Function for Probability of Utilization of Destination

Prospect Theory takes into account the tendency for individuals to weight small changes in probability at the extremes more than the equivalent numerical change in the middle of the probability scale. The actual curve of the prospect function can be affected by a number of factors; the person's natural risk bias, a person's experience in the area, consequences of the decision, the decision context, and more. The function given in Equation (2), as calibrated by Tversky and Kahneman, is used as the baseline for the GERBIL model and is repeated in Figure 7. The x-axis is the probability of an outcome while the y-axis is the weighted probability a person puts on the outcome, or in VFT terms the value placed on that probability. Harbaugh (2003) used the example of gambling to explain this phenomenon. For example, in the region of low probability on the scale, a person essentially weights the probability of success of a gamble as 0.156, even though the gamble has a 0.10 chance of winning. This is equivalent to the gambler

considering a certain \$15.60 payout (expected value (EV) = \$15.60) and a 0.10 chance at a \$100 outcome (EV = \$10) to be equal. This equates to a negative risk premium of \$5.60 and is viewed as risk seeking behavior. The gambler attaches a value of 0.156 on the probability of 0.1. On the other end of the scale, the same person would judge the probability of success of a gamble as 0.766, even though the gamble has an actual probability of success of 0.90. This is equivalent to the person considering a certain payout of \$76.60 (EV = \$76.60) and a 0.90 chance at a \$100 outcome (EV = \$90) to be equal. This equates to a positive risk premium of \$13.40 and is risk averse. In this case, the gambler attaches a value of 0.766 on the probability of 0.9. There is a point in the function where it changes from risk seeking to risk aversion, which using equation (2) occurs approximately at 0.34.

Due to the human nature of assigning value to an uncertain event, Prospect Theory is used in the model to arrive at a value weighting for the probability of an event happening in the region. On the low end of the scale, a DM is expected to display risk seeking characteristics and therefore overweight the value of the corresponding probability. In the middle and high end of the scale, the DM is expected to display increasingly risk averse characteristics and therefore underweight the value of the corresponding probability according to behavioral theory. This novel utilization of Prospect Theory in this area of decision making helps to account for some of the behavioral aspects when assigning a probability to an unknown event. In this case, the prospect function, $\pi(P_d)$, uses the probability elicited from the DM, P_d , to calculate the value, $V^P(A_d)$, for the probability of an event occurring in the destination region which would require the alternative to be utilized. In assigning a value to a probability in this situation, a DM would be inclined to ascribe disproportionate value to low probability events. An example of this type of behavior would be the desire by the DM to not be caught off guard by an event with a low

probability of occurrence. In the middle of the range of the prospect function, 0.2 to 0.7, the slope is relatively flat and the value only increases from about 0.25 to 0.50. In this region, a DM does not perceive a significant value differential that corresponds to a similar increase on value to a corresponding increase in probability. For example, the DM would place a near-equal value on an occurrence with a 0.6 probability as an occurrence with probability of 0.4. Finally, on the high end of the probability scale, the DM would tend to underestimate the value of occurrences which have a high probability of occurrence. This is because once an event is no longer certain in the mind of the DM, i.e., where $P_d = 1.0$, the value placed on that probability assessed becomes lower than the actual probability. This represents a disproportionate aversion as the event moves from certainty to uncertainty; that is, the DM instinctively hedges against the uncertainty.

A proper assessment of the probability of utilization for a destination must take into account the number of possible APODs which could be used to access a particular region. While some regions around the world may only have one destination airfield identified to serve as an APOD for a crisis, many regions have two or more airfields identified as APODs. In these cases where a destination region is serviced by more than one APOD, it is important to understand the probabilities for utilization of each destination airfield may not be equal. For instance, prior to Operation IRAQI FREEDOM, both Kuwait International and Ali Al Salem Kuwait could be considered as possible APODs for any crisis in Iraq. While both airfields are relatively close to each other, the probability of utilization for Kuwait International would be much larger than that for Ali Al Salem. This is because most crises could be handled by Kuwait International alone, without having to utilize Ali Al Salem except for the larger and less probable crises.

Note that destination airfields may not be required solely for military operations. As the tragic tsunami in the Indian Ocean and the earthquake in Pakistan recently demonstrated,

America's unsurpassed logistical capability, supported by the backbone of the ERS, may be called upon for humanitarian operations anywhere in the world at a moment's notice.

National Security Import of Destination Region

The third branch is a measure of the National Security Import in the destination region, measured using a scale of 1 to 10. The constructed, proxy chart in Table 2 is used as a guide. This constructed, proxy scale was developed by the authors to provide a framework for users to base judgments for National Security Import.

Table 2: Constructed, Proxy Scale for National Security Import

NATIONAL SECURITY IMPORT		Minor Allies	Major Allies	US
1	None. Region has no impact in any area for US or allies.	GREEN	GREEN	GREEN
2	Slight. Instability/crisis in region has minimal impact on some US minor allies, but not the US or its major allies.	YELLOW	GREEN	GREEN
3	Minimal. Instability/crisis in region has major impact on some minor allies or minimal impact on some major allies. No impact on the US.	ORANGE	YELLOW	GREEN
4	Low. Instability/crisis in region has significant impact on some minor allies and minimal impact on some major allies. No impact on the US.	RED	YELLOW	GREEN
5	Low/Moderate. Instability/crisis in region has significant impact on some minor allies. Minimal impact on major allies and the US.	RED	YELLOW	YELLOW
6	Moderate. Instability/Crisis has minimal impact on major allies and minimal impact on US.	RED	ORANGE	YELLOW
7	Moderate/High. Instability/Crisis has major impact on major allies and minimal impact on US.	RED	ORANGE	YELLOW
8	High. Instability/Crisis has major impact on major allies and major impact on US.	RED	ORANGE	ORANGE
9	Very High. Instability/Crisis has significant impact on major allies and major impact on US.	RED	RED	ORANGE
10	Extremely High. Instability/Crisis definitely has significant impact on US and major allies.	RED	RED	RED

GREEN – no impact; YELLOW – minimal impact; ORANGE – major impact; RED – significant impact

Let,

N_d	National Security Import ascribed to the region operationally served by destination d . $N_d = [1, 10]$.
$v(N_d)$	SDVF for the National Security Import of destination d .

The value function for National Security Import is then given by equation (4).

$$V^N(A_d) = \begin{cases} v(N_d) & d \in D_A, \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

This value function is monotonically increasing. As the National Security Import moves up the scale from 1 to 10, the DM places increasing value on National Security Import. In the scale from 1 to 4, there is no impact to the US and thus there is very little value placed on it. After 5 on the scale, however, the US begins to become affected by crises in the region and this causes the DM to place a quickly increasing value on National Security Import. This can be seen in the SDVF for National Security Import in Figure 8. The DM has the ability to input a value here either deterministically or stochastically. If the input is stochastic, it is evaluated as a triangular distribution as previously discussed in the tactical sub-model. Using Kuwait International again, one would expect a DM to give it a score of 9 or 10 since the global economy is so dependent upon the flow of oil from the region. In addition, the US has demonstrated its commitment to the region numerous times. On the other end of the scale, the region served by the destination of Lumbashi International, Congo may only be assessed a 2 or 3.

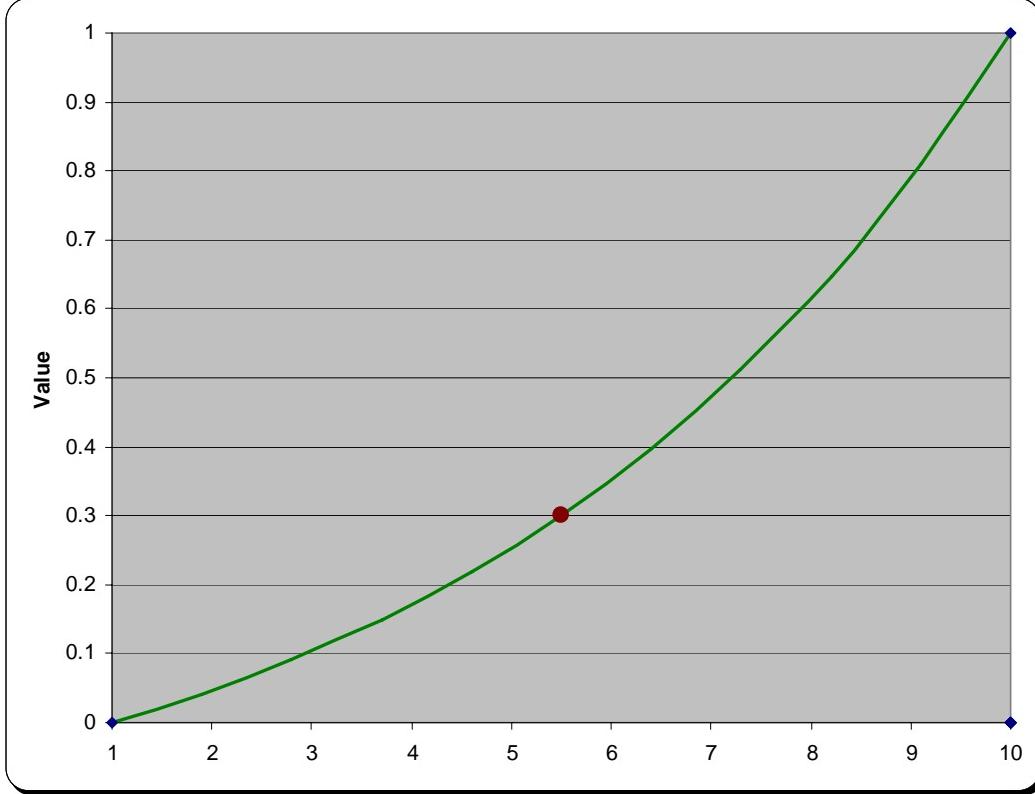


Figure 8: SDVF for National Security Import

Aggregation

Once values for all three branches are calculated, each branch is then weighted and summed as shown in equation 5 below to arrive at an en route alternative's overall value for a specific origin and destination.

$$V_{OP}(A_d) = \begin{cases} w_{OP}^T V^T(A_d) + w_{OP}^P V^P(A_d) + w_{OP}^N V^N(A_d) & d \in D_A, \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

$$\text{where } \sum_{i \in \{T, P, N\}} w_{OP}^i = 1$$

This process is repeated for an en route alternative for each of the entire set of destinations. This gives a total value, $V_{OP}(A)$, for a specific en route alternative given a fixed origin.

$$V_{OP}(A) = \sum_{d \in D} V_{OP}(A_d) = \sum_{d \in D_A} V_{OP}(A_d) \quad (6)$$

Observe that for destinations $d \in \overline{D}_A$ (i.e. destinations not feasible to A), that equation (5) is the null value. Hence, en route alternatives that are capable of servicing more destinations would, in general, receive higher total value scores as calculated by equation (6).

There are two main airlift hubs outside of CONUS; one at Ramstein AB, Germany and one at Yokota AB, Japan. Experiments were performed with Ramstein AB as an APOE to examine the extension of the Global Reach of the current ERS using a two-enroute-stop/three-flight profile. Operational values were summed for an alternative using Ramstein as an origin and using a wide range of predetermined destinations. The result is a total operational value for each en route airfield alternative for a given origin. These results provide insight to the DM about how different en route airfields rank for a specific region of the world.

After a short discussion on the user interface, the Global En Route Base Infrastructure Location Model (GERBIL), a case study using Ramstein AB, Germany as the origin is examined.

GERBIL

Set Up

GERBIL is an Excel based computer program built with Visual Basic Application 6.0 (VBA), providing users with a friendly interface to examine different scenarios. These scenarios assess the value of multiple alternative en route bases by automatically calculating how well each en route airfield alternative supports multiple destinations from a single origin.

GERBIL is set up to guide users through the process of entering the required data for origins, alternative en route airfields, and destinations for the value hierarchies. Only the major components will be discussed here; detailed programming and functionality, which includes an

integrated help function, are in Miravite and Schlegel (2006). Tables 3 and 4 show the input data requested for the origin and destination that is used in the case study discussed in the next section. Figure 9 shows the data input screen for en route airfield alternatives. Note the capability to input uncertainty into most measures.

Origin/Destination Airfields Data

Table 3: Origin Airfield Data

ORIGIN	Latitude			Longitude		
	N/S	Deg	MM.MM	E/W	Deg	MM.MM
Dover AFB, DE, USA (KDOV)	N	39	7.770	W	75	27.960
Fairford AB, UK, (EGVA)	N	51	40.90	W	1	47.40
Charleston, SC, USA (KCHS)	N	32	53.92	W	80	2.43
Mildenhall, England (EGUN)	N	52	21.72	E	0	29.18
Moron AB, SP (LEMO)	N	37	10.50	E	5	37.00
Rota NS, SP (LERT)	N	36	38.70	E	6	21.00
Spangdahlem AB (ETAD)	N	49	58.40	E	6	41.50
Ramstein AB, GE (ETAR)	N	49	26.20	E	7	36.00

Table 4: Destination Airfield Data

DESTINATION	Probability of Utilization	National Security			Latitude			Longitude		
		Min	Mode	Max	N/S	Deg	MM.MM	E/W	Deg	MM.MM
Hosea Kutako Intl, Namibia (FYWH)	0.20	2	3	4	S	22	28.79	E	17	28.26
Cape Town Intl, South Africa, (FACT)	0.10	2	3	5	S	33	57.90	E	18	36.10
Lubumbashi Intl, Congo (FZQA)	0.20	2	3	4	S	11	35.47	E	27	31.85
Lusaka International, Zambia (FLLS)	0.30	2	3	3	S	15	19.85	E	28	27.16
Dar Es Salaam, Tanzania (HTDA)	0.20	2	3	3	S	6	52.69	E	39	12.16
Chhatrapatii Shivaji International, India (VABB)	0.10	7	8	9	N	19	5.32	E	72	52.07
Diego Garcia NSF, British Indian Ocean Territory (FJDG)	1.00	8	8	9	S	7	18.78	E	72	24.65
Arkonam, India (VOAR)	0.10	7	8	9	N	13	4.27	E	79	41.47
Bandaranaike Int Columbol, Sri Lanka (VCBI)	0.05	4	5	6	N	7	10.87	E	79	53.07
Tribhuvan Intl, Nepal (VNKT)	0.05	3	4	5	N	27	41.78	E	85	21.55

Add a new airfield alternative

Airfield Name	[]				
Latitude	N <input type="button" value="deg"/> 0 <input type="button" value="deg"/> 0.00 <input type="button" value="mm.mm"/>	Longitude E <input type="button" value="deg"/> 0 <input type="button" value="deg"/> 0.00 <input type="button" value="mm.mm"/>			
Latitude and Longitude inputs will be converted to Degree Decimal format to be used in Great Circle Distance Calculations					
Assign Probability? <input type="checkbox"/> <input type="button" value="?"/>		Assign Probability? <input type="checkbox"/> <input type="button" value="?"/>			
MOG	0 <input type="button" value="?"/>	Seaport	0 <input type="button" value="?"/>	Mountainous	Yes <input type="button" value="?"/>
Critical Leg	Values automatically calculated on Alternatives page	Railroad	0 <input type="button" value="?"/>	Altitude	Yes <input type="button" value="?"/>
Delta Flight Length		Road System	0 <input type="button" value="?"/>	Weather	Heavy <input type="button" value="?"/>
Alt Airfields	0 <input type="button" value="?"/>	Commercial Airport	0 <input type="button" value="?"/>	Temperature	Yes <input type="button" value="?"/>
Fuel Storage	0 <input type="button" value="?"/>	Lodging	Inadequate <input type="button" value="?"/>	Urban Areas	Yes <input type="button" value="?"/>
Fuel Resupply	0 <input type="button" value="?"/>	Dining	Inadequate <input type="button" value="?"/>	Terrain	Yes <input type="button" value="?"/>
Diplomatic Clearance	None <input type="button" value="?"/>	Medical	No <input type="button" value="?"/>	 Show Measure Definitions	
Force Protection	Complete <input type="button" value="?"/>	Comm	None <input type="button" value="?"/>	<input type="button" value="Accept"/>	
Dept of State	Unacceptable <input type="button" value="?"/>	Power	None <input type="button" value="?"/>	<input type="button" value="Cancel"/>	
Military Cooperation	Poor <input type="button" value="?"/>	Water	None <input type="button" value="?"/>		
		Sewer	None <input type="button" value="?"/>		

Place mouse over input box to view the range for the measure inputs

Figure 9: GERBIL data input screen for en route alternatives

An added feature is the capability to change the default weights of both the tactical sub-model and the operational value hierarchy. One of the keys to a useful value hierarchy is assigning the correct weights for a specific DM. The weights of each objective can be changed locally for each branch, and the program includes an application of Clemen and Reilly's (2001) swing weighting technique to make the solicitation of weights from the DM easier. GERBIL displays a chart of the global weights for each objective and corresponding measures, demonstrating their contribution to the overall value score, shown in Figure 10.

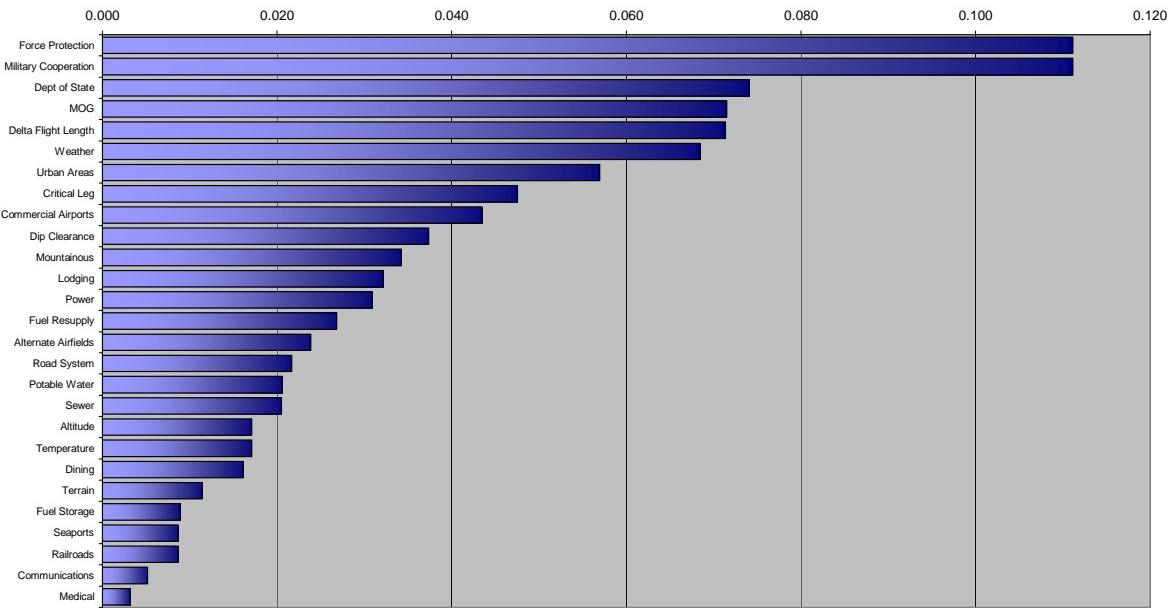


Figure 10: Global weights for tactical sub-model

The user can also change the underlying value function for a measure. This is useful if there is any change in requirements or capabilities, such as the introduction of a new strategic airlift aircraft having a smaller footprint. In this case, one of the value functions that might require adjustment is for MOG. If the new aircraft had the same cargo capability with a smaller footprint, the number of narrow body spots required for the same throughput would decrease. Thus the value function would be changed to reflect this new reality.

Once a user starts changing weights or value functions, GERBIL can save these new settings as the default settings. While any of the weights can be changed to give the model increased usefulness, the user can always go back to the original settings by using the restore function. This built in safety feature allows the user to feel confident in adapting weights and value functions, always knowing there is a way to get back to the original settings.

Case Study

Set Up

GERBIL gives the user the flexibility to run the tactical sub-model on its own with only a single origin and destination pair. For example, using Ramstein AB Germany as the origin and Lumbashi Intl Congo as the destination would yield the results in Figure 11. In this case the best alternative for an en route base is Dakar Senegal. Although Constanta, Romania scored better, it was not a valid option due to the critical leg being greater than the range of the C-17, 3,500 nm, and is flagged as such by the asterisk. GERBIL automatically screens out options which are not viable and labels them in the output, in this case with an asterisk. The constrained options are displayed to give the user additional insight should the constraining conditions change. For instance, if the government of Venezuela were to be replaced by a democratically elected government more favorable to the US, it is conceivable that we could establish an ERS in that country in the future. Due to the situation at present, however, any en route alternative in Venezuela would be automatically flagged for a constraint, based on inadequate host nation relations.

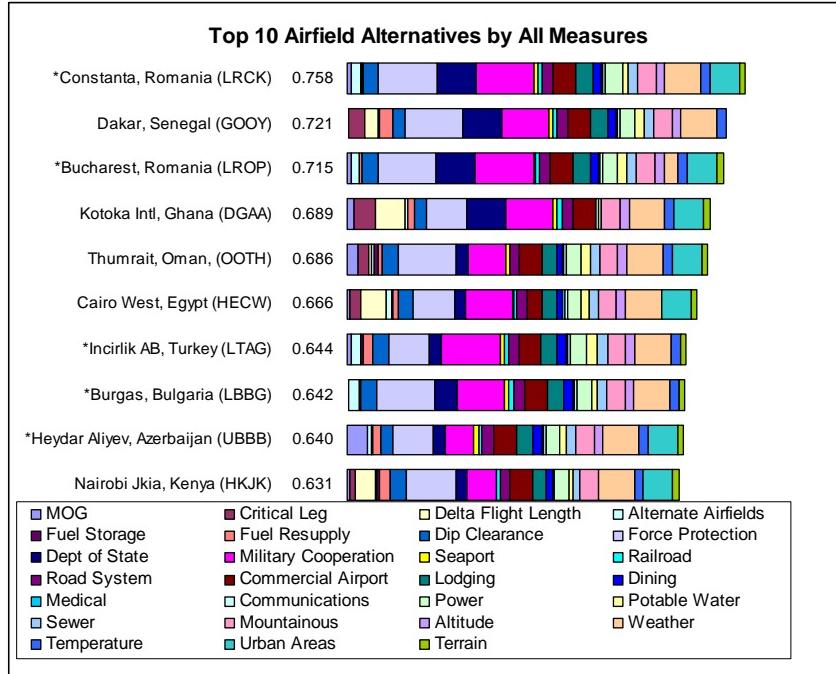


Figure 11: GERBIL results of tactical sub-model run, Ramstein AB Germany to Hoseo Kutako Intl Namibia

While the results of the tactical sub-model are very useful for providing insight on a particular origin and destination pairing, the utility of an alternative across a range of destinations provides more insight on a strategic level. In this case, the operational scenario is run from GERBIL. For this case study, Ramstein AB was selected as the APOE with ten APODs were spread across Africa (4) and Asia/Indian Ocean (5). The spread was intentionally uniform geographically so as not to favor one region over another. Each of these APODs was selected to be outside the current reach of Ramstein, thereby forcing a stop at an en route airfield. Fourteen en route airfield alternatives were identified to provide a sampling of possibilities to expand the current ERS from its current boundaries in different directions.

With GERBIL populated with case study data, the operational scenario was run with a weighting of 0.6 for the tactical sub-model, 0.2 for the probability of using the APOD, and 0.2 for the National Security import of the region. These weightings were chosen to give the tactical

sub-model portion three times the weighting of the measures for both the probability of using the APOD and the National Security Import. This also equally weights the measures for the probability of using the APOD and National Security Import. While these weightings were chosen by the authors for this scenario, GERBIL allows these weights to be easily modified by the user to fit their own values and scenarios. The results are shown in Figure 12, with Thumrait, Oman scoring the highest followed by Nairobi Jkia, Kenya. From the chart it is evident that the main factor in both of them scoring so highly was their ability to reach so many APODs, 9 and 10 respectively, relative to alternatives such as Burgas, Bulgaria (7 APODs) and Constanta, Romania (6 APODS). This makes sense, however, since if only one new airfield was to be added to the ERS, the more APODs it could access which are currently outside the reach of the ERS, the better. Table 5 gives more detail on the results of the run for each alternative and destination APOD. This allows a user to focus in on specific regions, thus quickly eliminating alternatives which score low in those regions.

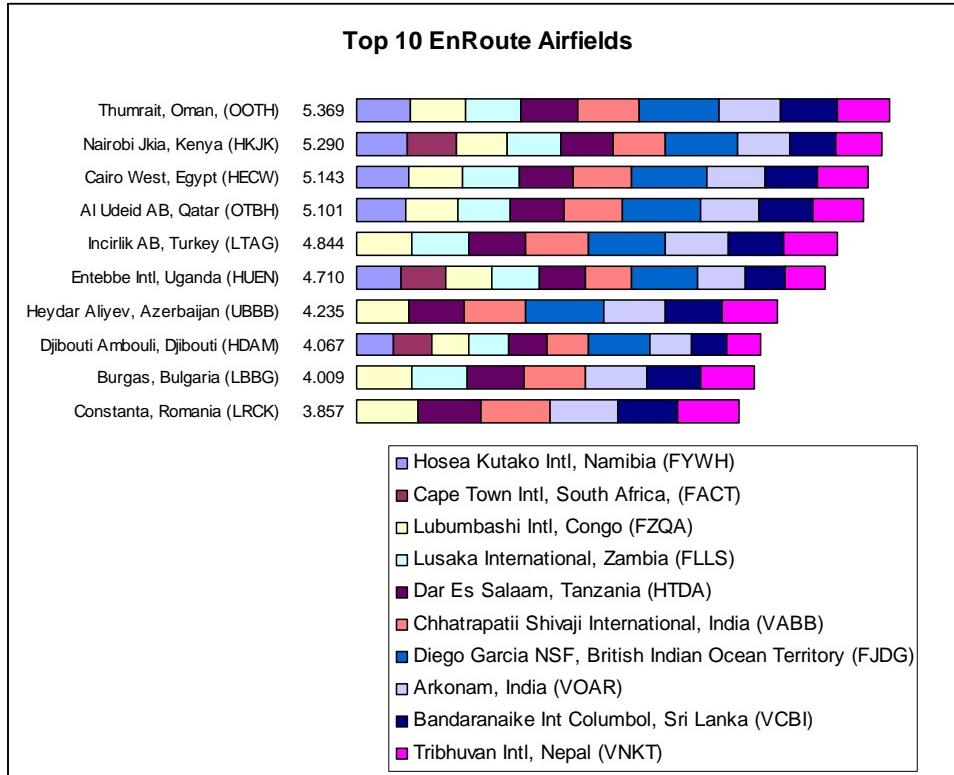


Figure 12: GERBIL results of operational scenario run with Ramstein AB Germany as APOE

Table 5: GERBIL results of operational scenario run with Ramstein AB Germany as the APOE

Rank	Alternatives	TOTAL	Hosea Kutako Intl, Namibia (FYWH)	Cape Town Intl, South Africa, (FACT)	Lubumbashi Intl, Congo (FZQA)	Lusaka International, Zambia (FLLS)	Dar Es Salaam, Tanzania (HTDA)	Chhatrapati Shivaji International, India (VABB)	Diego Garcia NSF, British Indian Ocean Territory (FJDG)	Arkonam, India (VOAR)	Bandaranaike Int Columbol, Sri Lanka (VCBI)	Tribhuvan Intl, Nepal (VNKT)
1	Thumrait, Oman, (OOTH)	5.369	0.540	0	0.553	0.567	0.564	0.617	0.808	0.620	0.566	0.534
2	Nairobi Jkia, Kenya (HKJK)	5.290	0.507	0.497	0.515	0.530	0.527	0.527	0.727	0.527	0.474	0.459
3	Cairo West, Egypt (HECW)	5.143	0.528	0	0.547	0.557	0.554	0.587	0.759	0.583	0.523	0.506
4	Al Udeid AB, Qatar (OTBH)	5.101	0.495	0	0.524	0.535	0.535	0.593	0.776	0.594	0.538	0.512
5	Incirlik AB, Turkey (LTAG)	4.844	0	0	0.562	0.570	0.574	0.625	0.785	0.621	0.559	0.549
6	Entebbe Intl, Uganda (HUEN)	4.710	0.455	0.438	0.464	0.477	0.468	0.469	0.667	0.469	0.414	0.389
7	Heydar Aliyev, Azerbaijan (UBBB)	4.235	0	0	0.529	0	0.551	0.625	0.786	0.624	0.564	0.556
8	Djibouti Ambouli, Djibouti (HDAM)	4.067	0.373	0.380	0.380	0.394	0.393	0.415	0.615	0.418	0.365	0.335
9	Burgas, Bulgaria (LBBG)	4.009	0	0	0.558	0.560	0.569	0.622	0	0.611	0.541	0.547
10	Constanta, Romania (LRCK)	3.857	0	0	0.620	0	0.634	0.692	0	0.681	0.611	0.619
11	Bucharest, Romania (LROP)	3.112	0	0	0.596	0	0.609	0.665	0	0.651	0	0.591
12	Bagram, Afghanistan (OAIK)	3.048	0	0	0	0	0.430	0.514	0.682	0.516	0.456	0.450
13	Kotoka Intl, Ghana (DGAA)	2.656	0.542	0.524	0.529	0.543	0.518	0	0	0	0	0
14	Dakar, Senegal (GOOY)	1.665	1.665	0.561	0	0.548	0.556	0	0	0	0	0

Once the operational scenario is complete, GERBIL has built in sensitivity analysis

functions. GERBIL provides the option of running sensitivity analysis on the weighting of any of the three branches of the operational value hierarchy. For example, when the weight of the tactical sub-model branch is changed in value between 0 and 1 for the sensitivity analysis, the

proportion of the weights between the other two branches to each other remains constant. Figure 13 shows the sensitivity analysis on the weighting for the tactical sub-model. When the weight of the tactical sub-model is decreased to approximately 0.46, Nairobi Jkia, Kenya replaces Thumrait, Oman as the preferred alternative. In the region above a weighting of 0.46 for the tactical sub-model, Thumrait dominates.

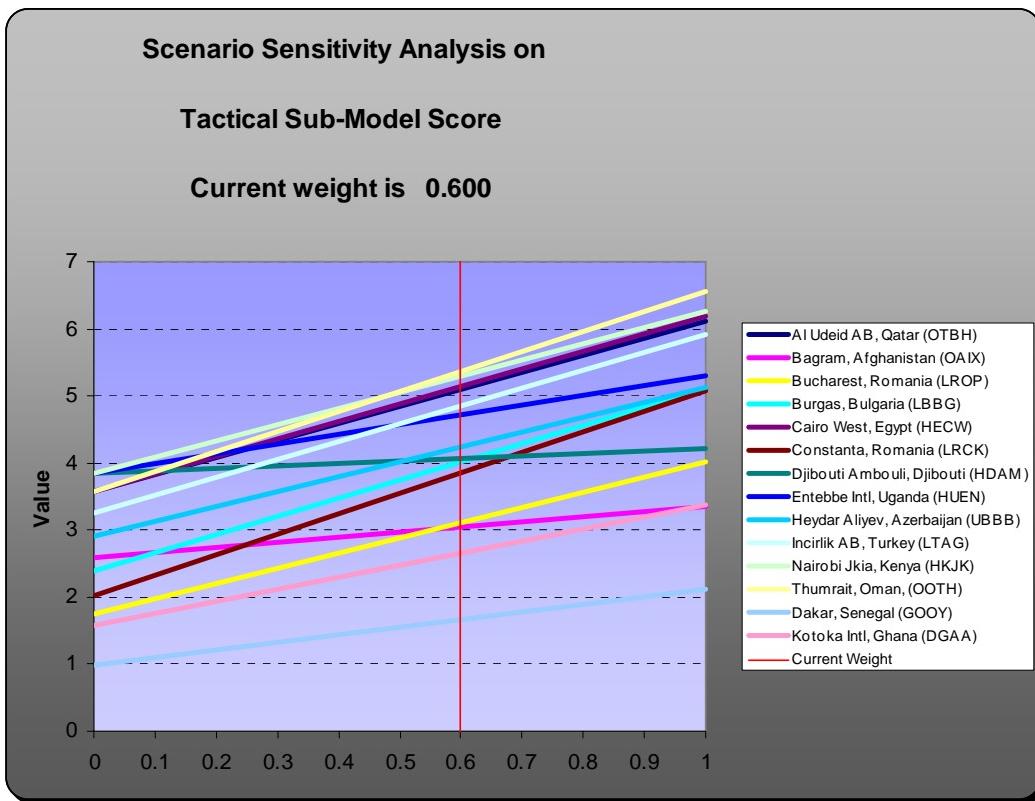


Figure 13: GERBIL sensitivity analysis

This same process of sensitivity analysis is also available when running just the tactical sub-model. The tactical sub-model allows for not only sensitivity analysis of any of the measures, but also for any of the higher-level objectives. See Miravite and Schlegel (2006) for detailed information on sensitivity analysis in the tactical sub-model.

Conclusions and Future Research

As the geopolitical climate continues to change, TRANSCOM must ensure a continuous capability to rapidly deliver cargo and personnel to wherever it is needed. The spreadsheet-based model and methodology developed in this paper provides an efficient means to analyze multiple en route alternatives to determine which is best suited for a particular region, given one origin and one destination. A further methodology is given to examine en route airfields that would add the most value for an entire operational region, given an origin airfield with multiple possible destinations spread out across a wide geographic area. Also, the addition of measures for both National Security Import and Probability of Utilization form a complete operational value hierarchy, ensuring that considered alternatives are thoroughly evaluated.

The key, however, was to put the methodology and mathematics of the underlying model into a user-friendly interface. GERBIL allows the user to input and remove origins, destinations, and en route airfield alternatives. SDVF_s and weights can be easily changed, with original or default values are easily restorable. GERBIL provides insight to the user to help make more informed decisions about locations which would add the most value to the ERS.

A future research would be to develop a means to incorporate cargo and passenger throughput requirements for destination regions. This would allow a more complete picture of the capabilities required to flow personnel and equipment through the system. The ability to evaluate multiple origins would provide an even greater strategic picture. Finally, this model assumes a “risk neutral” decision maker. The inclusion of risk tolerance parameters would further improve the model. While these areas would improve the power of the model, it is already capable of providing powerful insight to DMs and SMEs on possible areas to improve the ERS.

Acknowledgements

We would like to thank Dr James Moore and the funding we received from the Air Force Office of Scientific Research, grant QAF 185025203206. We would also like to thank Lt Col Diane Allen and TRANSCOM/J5 for sponsoring this line of mobility research.

Author Statement

The views expressed in this article are those of the authors and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the US Government.

II. Extended Narrative

Changes to the Tactical Sub-model

Background

The tactical submodel addressed in Section I is based off the value hierarchy developed by Capt Tharaldson for her 2006 AFIT thesis, *Strategic Airlift En Route Analysis to Support the Global War on Terrorism Using a Value Focused Thinking Approach*. In her thesis, she developed a value hierarchy to enable the analysis of different en route airfield alternatives for inclusion in the en route system (ERS). This original value hierarchy encompassed 27 measures which fell under two main objectives, throughput and sustainment. These 27 measures were evaluated for each en route airfield alternative for a specific origin airfield and destination airfield. The value scores are then used to determine the most preferred en route airfield alternative for the given pairing. While the original value hierarchy was a very useful product, some modifications had to be accomplished.

Critical Leg Measure

The first modification was on the single dimensional value function (SDVF) for the critical leg measure. The critical leg is defined as the longer leg of the two-leg flight; the distance from the origin airfield to the en route airfield alternative or the distance from the en route airfield alternative to the destination airfield. The critical leg can not exceed 3,500 nm as this is the maximum nominal range for a C-17 with a load of 45 short tons (McVicker, 2002). The original SDVF had this measure as a continuous increasing value function. This type of SDVF would give a higher value for a longer critical leg. What is wanted, however, is a SDVF which gives a higher value to a shorter critical leg. For example, if the distance between an

origin airfield and a destination airfield was 6,000 miles, you would want the critical leg to be 3,000 miles. This is exactly half the distance to be flown. If the two legs, however, were 2,500 miles and 3,500 miles, then the critical leg would be 3,500 miles. While this is still within the 3,500 nm range, it prohibits any cargo load above 45 short tons and does not give much room for unplanned deviations (i.e. weather diverts, air traffic control routing, etc...). If the critical leg was 3,000 nm though, more cargo could be carried and there would also be an extra margin of safety. For these reasons, the SDVF was changed from a continuous increasing to a continuous decreasing function as shown in Figure 4. The minimum number on the scale is 1,751 nm. This is because a critical leg score of 1,750 nm or less implies the total flight distance for both legs is 3,500 nm or less and an en route stop is therefore unnecessary. The maximum critical leg number is as discussed above, 3,500 nm.

Maximum On Ground (MOG)

There were two measures from the original value hierarchy which were consolidated into one measure, the parking MOG measure and the working MOG measure. For some of the airfields, the Airfield Suitability Reports from Air Mobility Command list different values for both the working MOG and the parking MOG. Normally, these values are not equal. For planning purposes, however, Air Mobility Command defined the parking MOG to be the specific number of aircraft which can be physically located on the ramp while the “working MOG is generally accepted to be one half the parking MOG (McVicker 2002).” Since it is essential for a value hierarchy to be mutually exclusive in its measures, one of the two original measures had to be eliminated. Therefore, the revised tactical submodel has only one measure for MOG. The GERBIL defines MOG for the tactical submodel as:

MOG counts the number of narrow body aircraft (C-17 equivalents) that can be parked, not receiving service, at an airfield at one time. These elements affect the throughput of a potential base by limiting the number of aircraft able to utilize an airfield on a given day. Working MOG is assumed to be half of the parking MOG per AMC guidance.

The SDVF for MOG is shown in Figure 14. The tactical submodel retained the same local and global weighting for MOG as in the original model.

Fuel Resupply

The original value hierarchy contained only one measure for fuel, fuel storage. The new tactical submodel added a second measure for fuel resupply, as shown in Figure 5, while keeping the measure for fuel storage. Fuel storage measures the amount of fuel stored, measured in millions of gallons, at an en route airfield alternative for airlift aircraft. This measure is essential to provide a steady supply of fuel throughout operations, even though the resupply may be affected by weather, maintenance, or other factors such as local holidays. In addition, the fuel storage provides a buffer for surge operations above the fuel resupply rate. The new measure for fuel resupply accounts for the total number of gallons, measured in millions of gallons, of aviation fuel which the en route airfield alternative has the capability to receive on a daily basis from all sources (i.e. pipelines and fuel trucks). It is essential for an en route airfield alternative to have a robust, fuel resupply capability to replenish the fuel stock as operations are sustained over a period of time. It should be obvious that measures for both fuel storage and fuel resupply at the en route airfield alternatives are required. An en route airfield alternative needs both the capability to store large amounts of fuel to weather supply disruptions and a healthy resupply of fuel to sustain daily operations for an extended period of time.

The creation of two measures for the Fuel objective required a new weighting assessment. The objective Fuel retained the same local and global weighting as in the original

model, while its two measures, Fuel Storage and Fuel Resupply, were assigned the weights shown in the Table below.

Table 6: Revised Fuel Weighting

Fuel		
	Fuel Storage	Fuel Resupply
Local Weights	0.250	0.750
Global Weights	0.009	0.027

Uncertainty

The biggest change found in the tactical submodel is the introduction of uncertainty. Previously, the inputs for all the measures were deterministic. While the user can still input a deterministic value into GERBIL, there are many instances where the answer is not known with certainty. In these situations, a stochastic input is appropriate. Two distributions were used for uncertainty, triangular and empirical.

The triangular distribution is commonly used in situations in which the exact form of the distribution is not known, but estimates (or guesses) for the minimum, maximum, and most likely values are available. The triangular distribution is easier to use and explain than other distributions that may be used in this situation (e.g., the beta distribution). (Kelton, 2004:632)

If the exact values are not known for an attribute that is measured continuously, the user can enter uncertain values using a triangular distribution.

- a_i^* measurement of the attribute i using a continuous triangular distribution
- a_{ia}^* minimum measurement of the uncertain attribute i
- a_{im}^* most likely (or mode) measurement of the uncertain attribute i
- a_{ib}^* maximum measurement of the uncertain attribute i

The expected value of a triangular distribution is given by:

$$E[x] = (a + m + b)/3$$

Therefore the expected measurement level of an uncertain continuous attribute i is given by:

$$E[a_i^*] = \frac{a_{ia}^* + a_{im}^* + a_{ib}^*}{3}$$

The expected value for the SDVF of attribute i is then

$$E[v_i(a_i^*)] = v_i(E[a_i^*]) = v_i\left(\frac{a_{ia}^* + a_{im}^* + a_{ib}^*}{3}\right) = \frac{1}{3}(v_i(a_{ia}^*) + v_i(a_{im}^*) + v_i(a_{ib}^*))$$

$$E[V(x)] = V(E[x])$$

A number of the attributes in the tactical submodel, however, are not measured continuously but rather categorically. If the categorical value for an attribute is not known with certainty, the user can enter a discrete empirical distribution specifying the probabilities that the attribute takes on the corresponding categorical values.

$a_i^\#$ measurement of the attribute i using a discrete empirical distribution

If an attribute can take on any of L categorical values, then the probability for each possible category is specified by the user.

$a_{iL}^\#$ categorical value L of the uncertain attribute i .

p_{iL} probability that attribute i takes on the categorical value L

The expected value of a discrete distribution is given by:

$$E[x] = \sum_i p_i x_i$$

In this case, the expected categorical measurement for attribute i is given by

$$E[a_i^\#] = \sum_L p_{iL} a_{iL}^\#$$

The expected value for the SDVF of an uncertain categorical attribute i is then

$$E[v_i(a_i^\#)] = v_i(E[a_i^\#]) = v_i\left(\sum_L p_{iL} a_{iL}^\#\right) = \sum_L p_{iL} v_i(a_{iL}^\#)$$

III. Future Research

Throughput

The first area of future research mentioned in Section I is to add throughput required for the destination airfield into the model. Due to the many, varied types of operations which could occur around the world, each destination region would have its own specific throughput given the most likely scenario in the region. For instance, a destination area in South Korea would require a significant amount of throughput since there is a possibility of a Major Theater War on the peninsula. The Philippines as the destination region, however, would require a much lower throughput. This is because their government is stable and they have no major enemies which would result in a major war in their country. A certain amount of throughput is required, however, for stabilization and anti-terrorist efforts in the region. In addition, a limited throughput for the region may be required for humanitarian efforts after a tsunami, typhoon, volcanic eruption, or other natural disaster. By accounting for these variances of throughput required in different destination regions, a better decision can be made on what is the best way to expand the ERS.

Multiple Origins

The second area for future research would be in providing for a methodology to roll up the value of an en route alternative airfield based upon not only multiple destination airfields but also multiple origin airfields. By modeling multiple origin airfields, it would better reflect what happens many times in the ERS. For instance, while a large part of the cargo bound for Kuwait during the lead up to OIF started their strategic flight from Charleston AFB SC, many flights went out of other locations such as Dover AFB DE. By allowing multiple origins, an en route

airfield alternative could be better judged on its ability to support operations which have their start at multiple origin airfields.

Risk Tolerance

The third area of possible future research would be to introduce risk into the model. A decision maker (DM) can be risk averse, risk neutral or risk seeking when making a decision. It is even possible for the same DM to be risk seeking when making a decision under certain circumstances and risk averse when making the same decision under different circumstances. When discussing risk tolerance and multiattribute objectives, the process of accounting for risk tolerance becomes even more complex.

The GERBIL model was developed as risk neutral. This was done for two reasons. The first was it simplified the process of analysis since a risk tolerance assessment would not need to be accomplished for each DM. In the case of GERBIL, the end DM might not even be known until well after the analysis was underway and results presented.

The second reason for a risk neutral stance is given by Kirkwood (1997) when he states that in only a small number of decisions, 5% to 10%, is the final decision affected by ρ_m , the multiattribute risk tolerance. In addition, ρ_m is almost always a positive number, reflecting the risk aversion bias of many DMs. Therefore, Kirkwood (1997) states that if values for ρ_m from 0.2 to infinity do not affect which alternative is preferred, then the multiattribute risk tolerance does not have to be considered and a value for ρ_m does not need to be computed.

After running the case study in Section I, the results were analyzed for $\rho_m = 0.2$, $\rho_m = 1$, $\rho_m = 10$, $\rho_m = 10000$. The original results were scaled to arrive at a value of between 0 and 1 for each alternative. This was accomplished by dividing their final values by the total number of

destinations considered, in this case 10. These results are shown in Tables 7 through 10. As can be seen, the preferred alternative of Thumrait, Oman (OOTH) remains the preferred alternative throughout the range of ρ_m . Therefore, in this case the preferred alternative does not change and hence the decision was made to not consider the multiattribute risk tolerance. Other case studies, however, may be affected and it would be an interesting area for future research.

Table 7: Multiattribute Risk Tolerance, $\rho_m = 0.2$

$\rho_m =$	0.2		
Alternative	Prelim score	Prelim Score Scaled	Risk Adjusted Column C
OOTH	5.369	0.5369	0.9381
HKJK	5.290	0.5290	0.9353
HECW	5.143	0.5143	0.9298
OTBH	5.101	0.5101	0.9282
LTAG	4.844	0.4844	0.9174
HUEN	4.710	0.4710	0.9113
UBBB	4.235	0.4235	0.8856
HDAM	4.067	0.4067	0.8750
LBBG	4.009	0.4009	0.8711
LRCK	3.857	0.3857	0.8604
LROP	3.112	0.3112	0.7944
OAIX	3.048	0.3048	0.7875
DGAA	2.656	0.2656	0.7399
GOOY	1.665	0.1665	0.5689

Table 8: Multiattribute Risk Tolerance, $\rho_m = 1$

$\rho_m =$	1		
Alternative	Prelim score	Prelim Score Scaled	Risk Adjusted Column C
OOTH	5.369	0.5369	0.6573
HKJK	5.290	0.5290	0.6499
HECW	5.143	0.5143	0.6361
OTBH	5.101	0.5101	0.6321
LTAG	4.844	0.4844	0.6074
HUEN	4.710	0.4710	0.5943
UBBB	4.235	0.4235	0.5461
HDAM	4.067	0.4067	0.5287
LBBG	4.009	0.4009	0.5224
LRCK	3.857	0.3857	0.5063
LROP	3.112	0.3112	0.4231
OAIX	3.048	0.3048	0.4156
DGAA	2.656	0.2656	0.3690
GOOY	1.665	0.1665	0.2427

Table 9: Multiattribute Risk Tolerance, $\rho_m = 10$ $\rho_m =$

10

Alternative	Prelim score	Prelim Score Scaled	Risk Adjusted Column C
OOTH	5.369	0.5369	0.5493
HKJK	5.290	0.5290	0.5415
HECW	5.143	0.5143	0.5268
OTBH	5.101	0.5101	0.5226
LTAG	4.844	0.4844	0.4969
HUEN	4.710	0.4710	0.4835
UBBB	4.235	0.4235	0.4357
HDAM	4.067	0.4067	0.4188
LBBG	4.009	0.4009	0.4129
LRCK	3.857	0.3857	0.3976
LROP	3.112	0.3112	0.3220
OAIX	3.048	0.3048	0.3155
DGAA	2.656	0.2656	0.2754
GOOY	1.665	0.1665	0.1735

Table 10: Multiattribute Risk Tolerance, $\rho_m = 10000$ $\rho_m =$

10000

Alternative	Prelim score	Prelim Score Scaled	Risk Adjusted Column C
OOTH	5.369	0.5369	0.5369
HKJK	5.290	0.5290	0.5290
HECW	5.143	0.5143	0.5143
OTBH	5.101	0.5101	0.5101
LTAG	4.844	0.4844	0.4844
HUEN	4.710	0.4710	0.4711
UBBB	4.235	0.4235	0.4235
HDAM	4.067	0.4067	0.4068
LBBG	4.009	0.4009	0.4009
LRCK	3.857	0.3857	0.3857
LROP	3.112	0.3112	0.3113
OAIX	3.048	0.3048	0.3048
DGAA	2.656	0.2656	0.2656
GOOY	1.665	0.1665	0.1665

Three Leg Flights, Two En Route Stops

This final area of future research was specifically requested by Lt Col Diane Allen in TRANSCOM/J5. Currently, GERBIL only allows for analysis of a two-leg flight, from an origin airfield to an en route alternative airfield and then on to a final destination airfield. While this limitation is overcome in GERBIL by specifying a current en route base, such as Ramstein AB, Germany, as the origin airfield, this is not ideal. By allowing a future model to make two en

route stops, a more realistic flow of materials and personnel from CONUS can be analyzed across a broader range of possibilities.

For instance, consider a scenario requiring Cape Town, South Africa as the final destination. In the current model, Ramstein is used as the origin and only en route alternatives located somewhere in Northern Africa or the Mid East are feasible as the ‘second en route stop’. There is another way, however, to get to Cape Town without transiting Europe. This is to utilize an en route stop somewhere in South America and then a second en route somewhere in Africa. Since the current model forces the first en route stop to be chosen as the origin, it sometimes does not allow direct comparisons of alternatives transiting different regions. The development of a future model to allow the capability of two en route stops would enhance the value of the current model for TRANSCOM/J5.

Appendix 1. Operational Model Data

Origin/Destination Airfields Data

Table 11: Origin Airfield Data

ORIGIN	N/S	Latitude		Longitude	
		Deg	MM.MM	E/W	Deg
Dover AFB, DE, USA (KDOV)	N	39	7.770	W	75
Fairford AB, UK, (EGVA)	N	51	40.90	W	1
Charleston, SC, USA (KCHS)	N	32	53.92	W	80
Mildenhall, England (EGUN)	N	52	21.72	E	0
Moron AB, SP (LEMO)	N	37	10.50	E	5
Rota NS, SP (LERT)	N	36	38.70	E	6
Spangdahlem AB (ETAD)	N	49	58.40	E	6
Ramstein AB, GE (ETAR)	N	49	26.20	E	7

Table 12: Destination Airfield Data

DESTINATION	Probability of Utilization	National Security			Latitude		Longitude		
		Min	Mode	Max	N/S	Deg	MM.MM	E/W	Deg
Hosea Kutako Intl, Namibia (FYWH)	0.20	2	3	4	S	22	28.79	E	17
Cape Town Intl, South Africa, (FACT)	0.10	2	3	5	S	33	57.90	E	18
Lubumbashi Intl, Congo (FZQA)	0.20	2	3	4	S	11	35.47	E	27
Lusaka International, Zambia (FLLS)	0.30	2	3	3	S	15	19.85	E	28
Dar Es Salaam, Tanzania (HTDA)	0.20	2	3	3	S	6	52.69	E	39
Chhatrapatii Shivaji International, India (VABB)	0.10	7	8	9	N	19	5.32	E	72
Diego Garcia NSF, British Indian Ocean Territory (FJDG)	1.00	8	8	9	S	7	18.78	E	72
Arkonam, India (VOAR)	0.10	7	8	9	N	13	4.27	E	79
Bandaranaike Int Columbol, Sri Lanka (VCBI)	0.05	4	5	6	N	7	10.87	E	79
Tribhuvan Intl, Nepal (VNKT)	0.05	3	4	5	N	27	41.78	E	85

Enroute Alternatives Data

Table 13: Enroute Alternatives Data Table A

Name (ICAO)	N/S	Latitude (decimal degree)	E/W	Longitude (decimal degree)	MOG	Critical Leg	Delta Flight Length	Alternate Airfields
AI Udeid AB, Qatar (OTBH)	N	25.117	E	51.315	9	2497	780	69
Bagram, Afghanistan (OAIX)	N	34.946	E	69.265	(3, 4, 5)	3035	2064	(30, 45, 55)
Bucharest, Romania (LROP)	N	44.571	E	26.085	(5, 5, 6)	3171	215	(80, 100, 125)
Burgas, Bulgaria (LBBG)	N	42.570	E	27.515	2	3038	197	128
Cairo West, Egypt (HECW)	N	30.116	E	30.916	(4, 4, 6)	2273	78	(40, 50, 60)
Constanta, Romania (LRCK)	N	44.362	E	28.488	(3, 5, 8)	3132	273	133
Dakar, Senegal (GOOY)	N	14.740	W	17.490	3	3617	9999	10
Djibouti Ambouli, Djibouti (HDAM)	N	11.547	E	43.160	(2, 3, 5)	2877	244	(5, 10, 25)
Entebbe Intl, Uganda (HUEN)	N	0.042	E	32.444	3	3231	46	14
Heydar Aliyev, Azerbaijan (UBBB)	N	40.468	E	50.047	(18, 20, 24)	2906	997	(20, 30, 40)
Incirlik AB, Turkey (LTAG)	N	37.002	E	35.426	(4, 7, 9)	2644	295	108
Kotoka Intl, Ghana (DGAA)	N	5.605	W	0.167	9	2662	1373	18
Nairobi Jkia, Kenya (HKJK)	S	1.319	E	36.928	(4, 5, 6)	3405	0	(8, 10, 17)
Thumrait, Oman, (OOTH)	N	17.666	E	54.025	13	2936	887	24

Table 14: Enroute Alternatives Data Table B

(ICAO)	Fuel Storage	Fuel Resupply	Dip Clearance	Force Protection	Dept of State	Military Cooperation
(OTBH)	2.2648	(0.708, 1, 1.1)	Limited	Moderate	Satisfactory	Good
(OAIX)	(0.5, 0.7, 1)	(0.12, 0.2, 0.3)	Limited	Complete	Good	Good
(LROP)	0.66	0.32	Limited	Minimal	Excellent	Excellent
(LBBG)	2.11	0.19	Limited	Minimal	Good	Good
(HECW)	(1, 2, 2)	(0.2, 0.5, 0.7)	Limited	Moderate	Satisfactory	Good
(LRCK)	1	0.25	Limited	Minimal	Excellent	Excellent
(GOOY)	0.793	1.43	Mission by Mission	Minimal	Excellent	Good
(HDAM)	0.354	(0.15, 0.354, 0.708)	Mission by Mission	Complete	Satisfactory	Fair
(HUEN)	0.634	0.14	Mission by Mission	Moderate	Satisfactory	Fair
(UBBB)	2.46	0.65	Mission by Mission	Moderate	Satisfactory	Fair
(LTAG)	2.26	(0.708, 1, 1.2)	Limited	Moderate	Satisfactory	Excellent
(DGAA)	0.264	0.608	Mission by Mission	Moderate	Excellent	Good
(HKJK)	2.64	1.14	Limited	(Moderate 0.5, Minimal 0.5)	Satisfactory	Fair
(OOTH)	5.2	0.39	Limited	Minimal	Satisfactory	(Good 0.5, Fair 0.5)

Table 15: En Route Alternatives Data Table C

(ICAO)	Seaport	Railroad	Road System	Commercial Airport	Lodging	Dining	Medical
(OTBH)	18	150	4	19	Adequate	Adequate	Yes
(OAI)	740	250	20	26	Partial	Partial	Yes
(LROP)	129	2	1	0	Adequate	Adequate	Yes
(LBBG)	2	1	1	0	Adequate	Adequate	Yes
(HECW)	95	8	3	30	(Adequate 0.2, Partial 0.8)	(Adequate 0.2, Partial 0.8)	Yes
(LRCK)	8	2	2	0	Adequate	Adequate	Yes
(GOOY)	0	4	1	0	Adequate	Adequate	Yes
(HDAM)	5	2	1	0	Inadequate	Inadequate	No
(HUE)	0	20	5	0	Adequate	Adequate	Yes
(UBBB)	10	2	1	0	Adequate	Adequate	Yes
(LTAG)	0	1	1	8	Adequate	Adequate	Yes
(DGAA)	0	0	0	0	Inadequate	Inadequate	No
(HKJK)	270	4	4	0	Partial	Partial	Yes
(OOT)	45	999	2	0	Partial	Partial	Yes

Table 16: En Route Alternatives Data Table D

Name (ICAO)	Communications	Power	Potable Water	Sewer
Al Udeid AB, Qatar (OTBH)	Robust	Robust	Adequate	Adequate
Bagram, Afghanistan (OAI)	Adequate	Adequate	Adequate	Adequate
Bucharest, Romania (LROP)	Robust	Adequate	Adequate	Adequate
Burgas, Bulgaria (LBBG)	Adequate	Adequate	Inadequate	Adequate
Cairo West, Egypt (HECW)	Adequate	Adequate	(Adequate 0.5, Inadequate 0.5)	(Robust 0.333, Adequate 0.333, Inadequate 0.333)
Constanta, Romania (LRCK)	Robust	Robust	Inadequate	Adequate
Dakar, Senegal (GOOY)	Adequate	Adequate	Adequate	Adequate
Djibouti Ambouli, Djibouti (HDAM)	Inadequate	Inadequate	Inadequate	Inadequate
Entebbe Intl, Uganda (HUE)	Inadequate	Inadequate	Inadequate	Inadequate
Heydar Aliyev, Azerbaijan (UBBB)	Adequate	Adequate	Inadequate	Adequate
Incirlik AB, Turkey (LTAG)	Robust	Robust	Robust	Robust
Kotoka Intl, Ghana (DGAA)	Adequate	Inadequate	None	None
Nairobi Jkia, Kenya (HKJK)	Inadequate	Adequate	(Inadequate 0.6, None 0.4)	(Adequate 0.35, Inadequate 0.65)
Thumrait, Oman, (OOT)	Adequate	Adequate	Adequate	Adequate

Table 17: En Route Alternatives Data Table E

Name (ICAO)	Mountainous	Altitude	Weather	Temperature	Urban Areas	Terrain
Al Udeid AB, Qatar (OTBH)	No	No	Average	Yes	No	No
Bagram, Afghanistan (OAIX)	Yes	Yes	Minimal	No	No	Yes
Bucharest, Romania (LROP)	No	No	Average	No	No	No
Burgas, Bulgaria (LBBG)	No	No	Minimal	No	Yes	No
Cairo West, Egypt (HECW)	No	No	Minimal	Yes	No	No
Constanta, Romania (LRCK)	No	No	Minimal	No	No	No
Dakar, Senegal (GOOY)	No	No	Minimal	No	Yes	Yes
Djibouti Ambouli, Djibouti (HDAM)	No	No	Minimal	Yes	Yes	Yes
Entebbe Intl, Uganda (HUEN)	No	Yes	Average	No	No	Yes
Heydar Aliyev, Azerbaijan (UBBB)	No	No	Minimal	No	No	No
Incirlik AB, Turkey (LTAG)	No	No	Minimal	No	Yes	No
Kotoka Intl, Ghana (DGAA)	No	No	Minimal	No	No	No
Nairobi Jkia, Kenya (HKJK)	No	Yes	Minimal	No	No	No
Thumrait, Oman, (OOTH)	No	No	Minimal	No	No	No

Tactical Submodel Value Functions

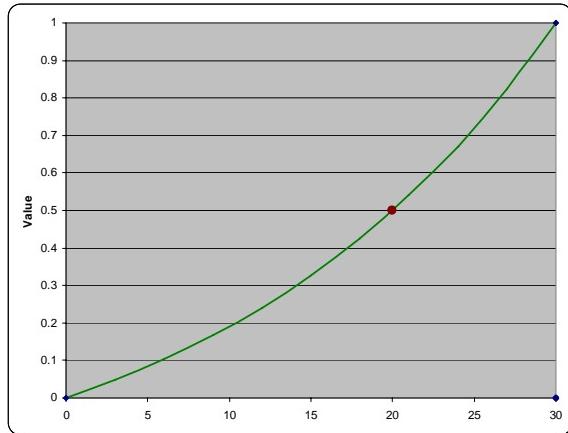


Figure 14: MOG SDVF

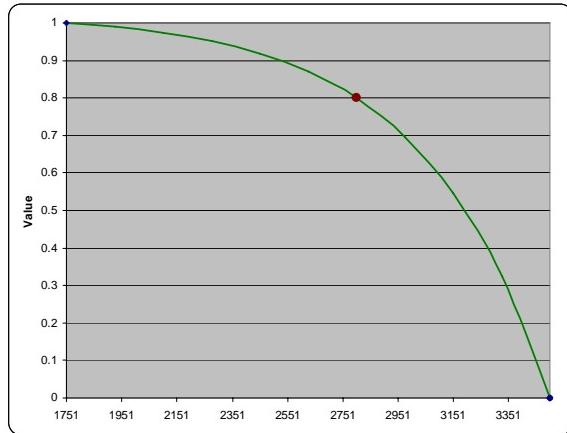


Figure 15: Critical Leg SDVF

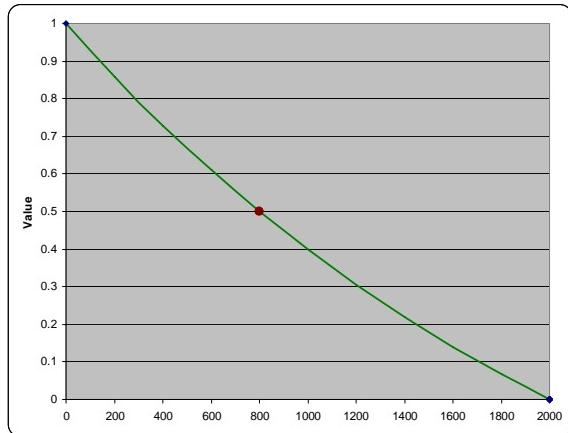


Figure 16: Delta Flight Length SDVF

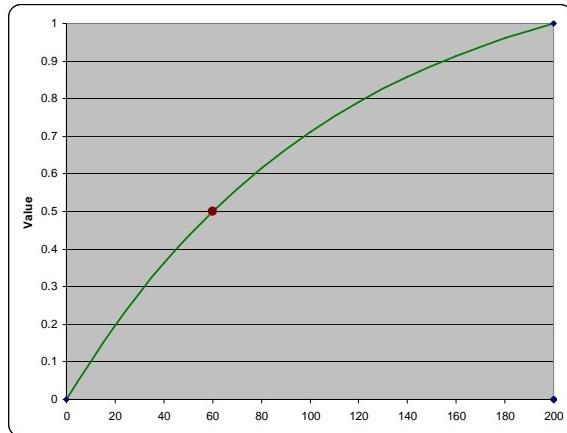


Figure 17: Alternate Airfields SDVF

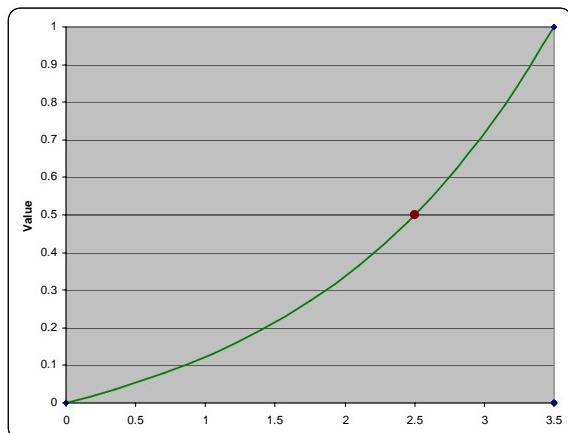


Figure 18: Fuel Storage SDVF

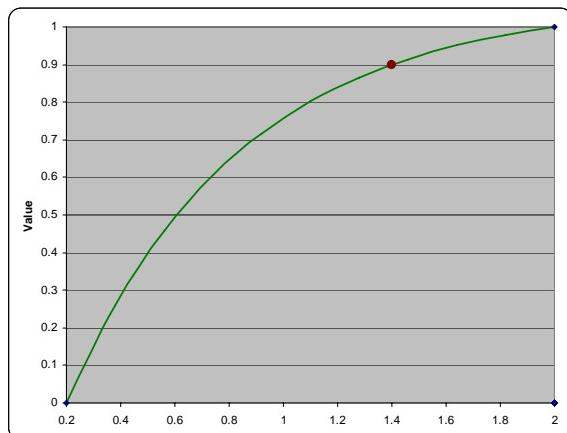


Figure 19: Fuel Resupply SDVF

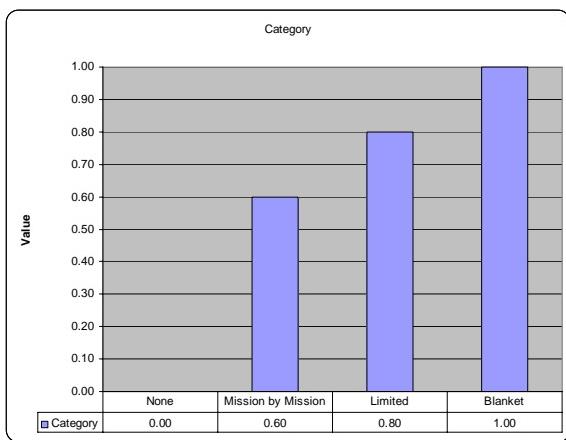


Figure 20: Diplomatic Clearance SDVF

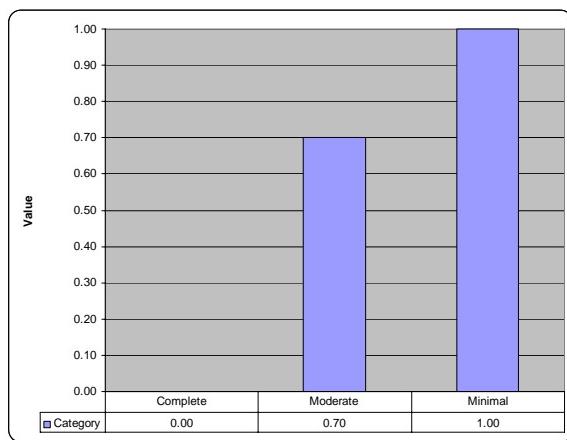


Figure 21: Force Protection SDVF

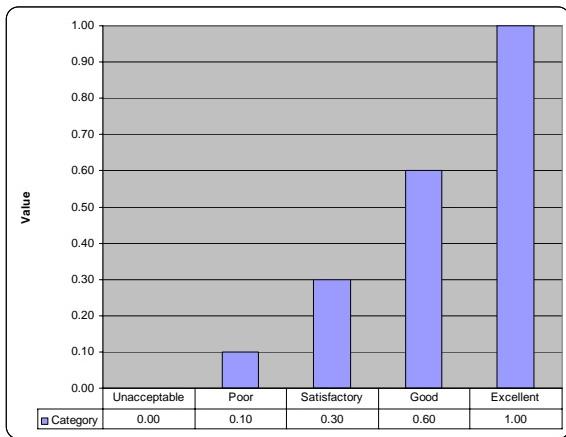


Figure 22: Department of State SDVF

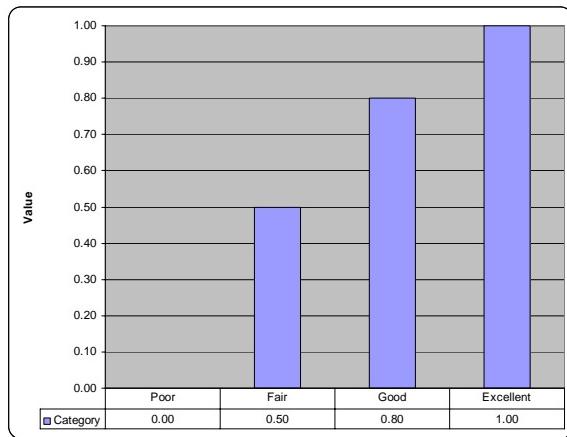


Figure 23: Military Cooperation SDVF

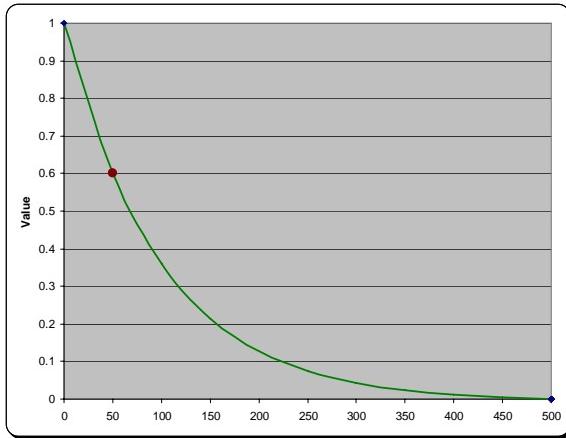


Figure 24: Seaport SDVF

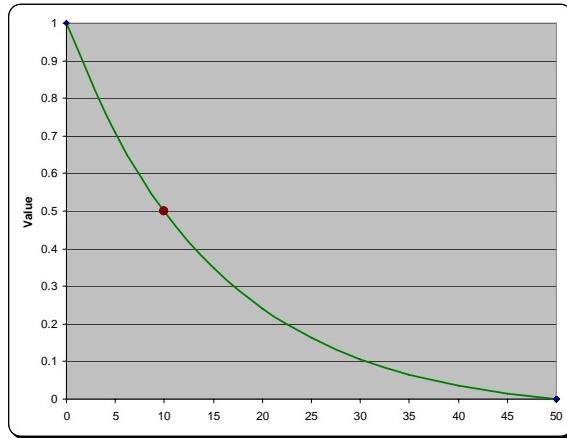


Figure 25: Railroad SDVF

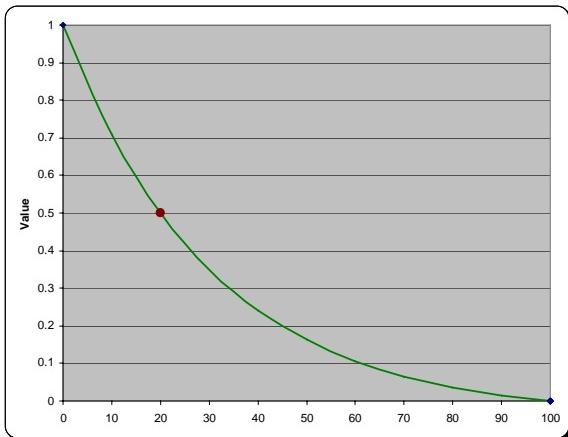


Figure 26: Road System SDVF

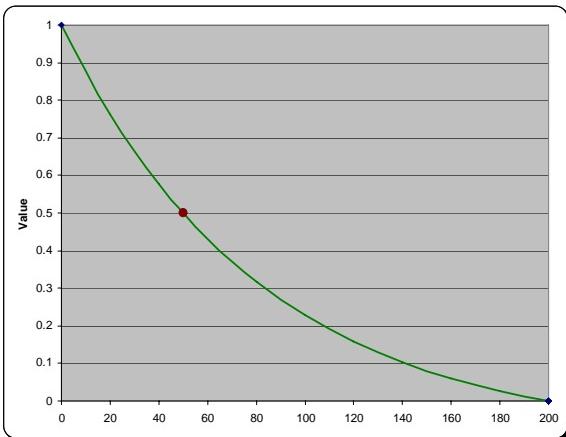


Figure 27: Commercial Airport SDVF

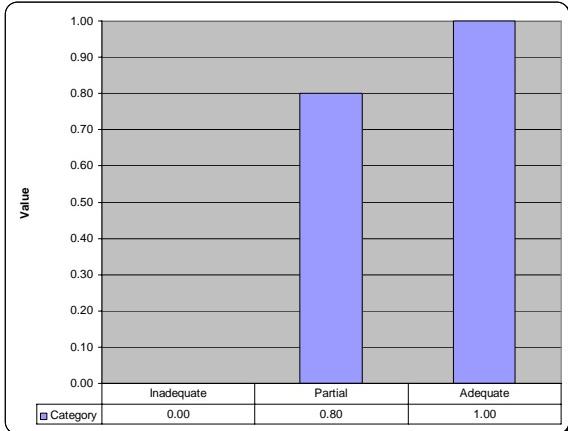


Figure 28: Lodging SDVF

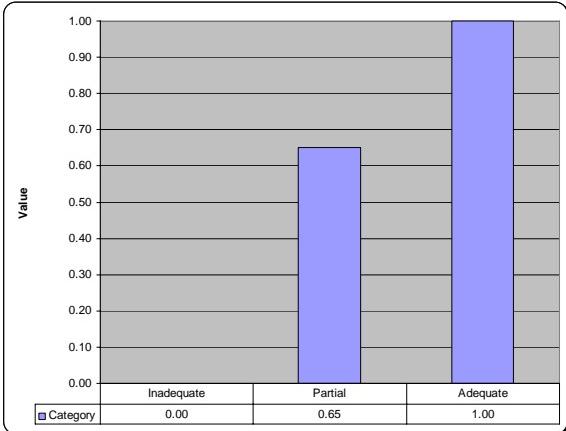


Figure 29: Dining SDVF

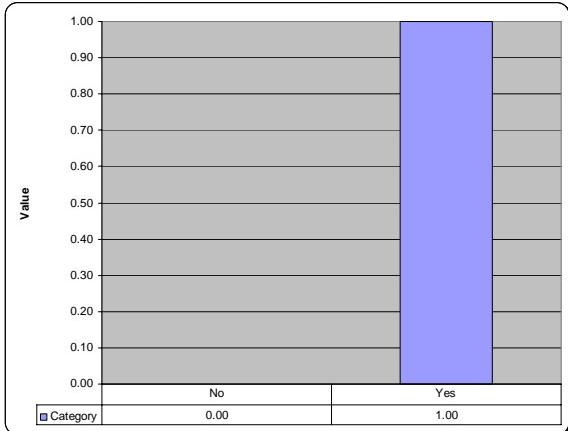


Figure 30: Medical SDVF

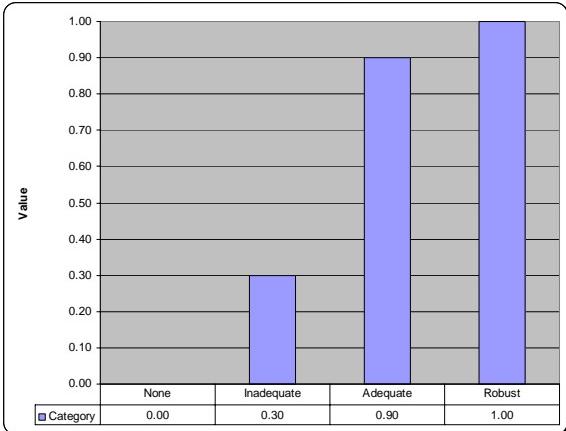


Figure 31: Communications SDVF

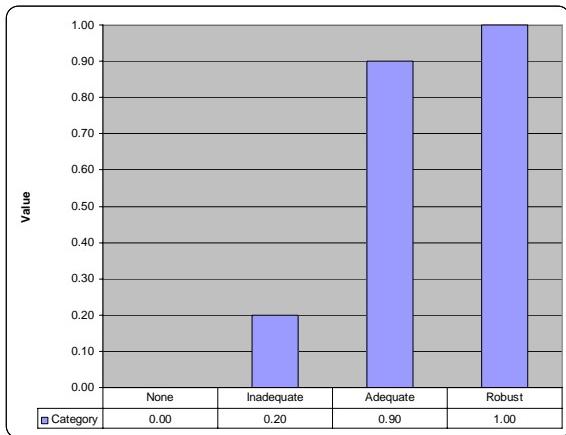


Figure 32: Power SDVF

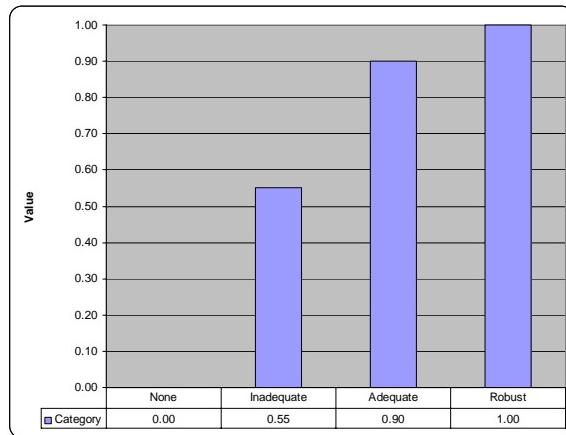


Figure 33: Potable Water SDVF

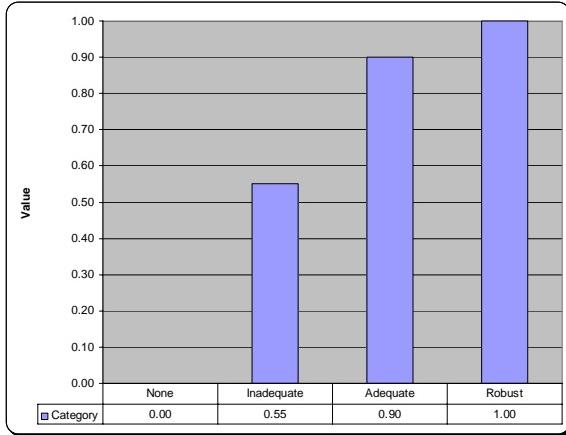


Figure 34: Sewer SDVF

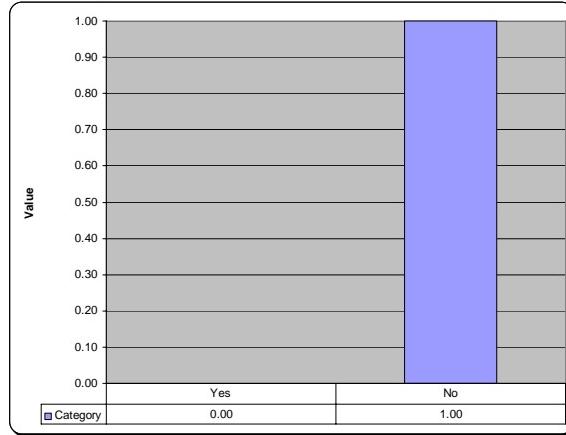


Figure 35: Mountainous SDVF

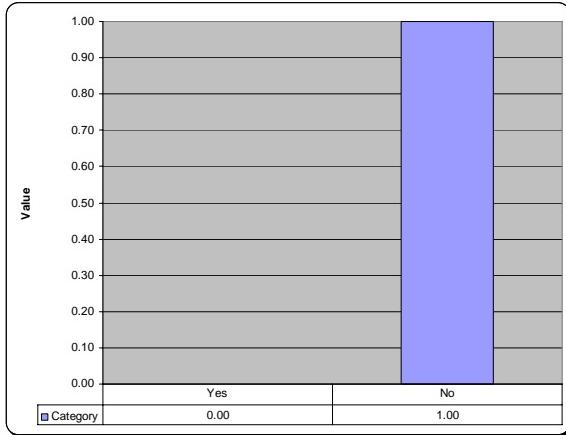


Figure 36: Altitude SDVF

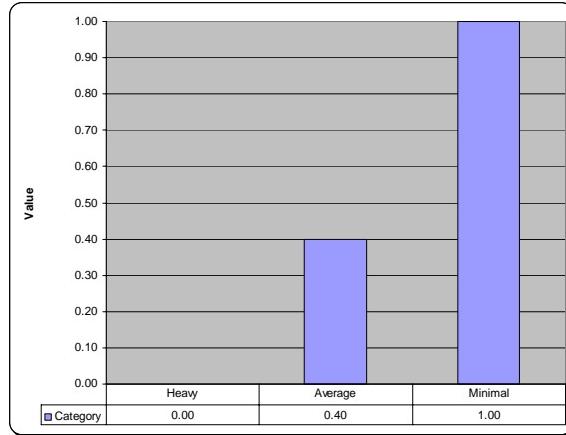


Figure 37: Weather SDVF

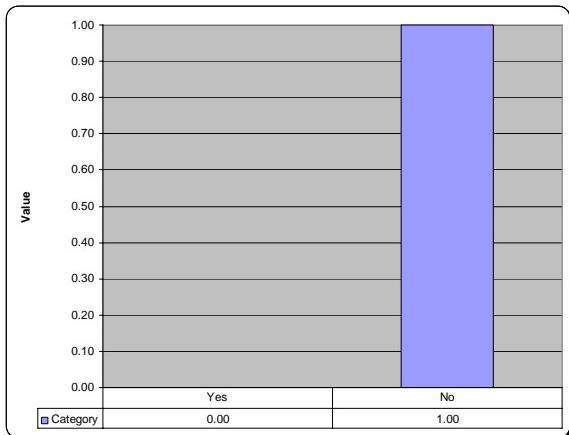


Figure 38: Temperature SDVF

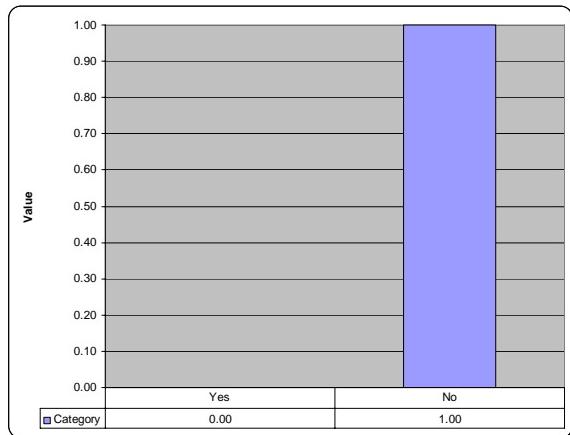


Figure 39: Urban Areas SDVF

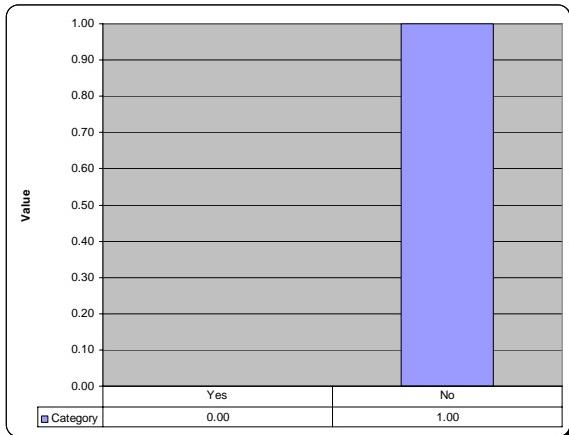


Figure 40: Terrain SDVF

Tactical Submodel Weighting

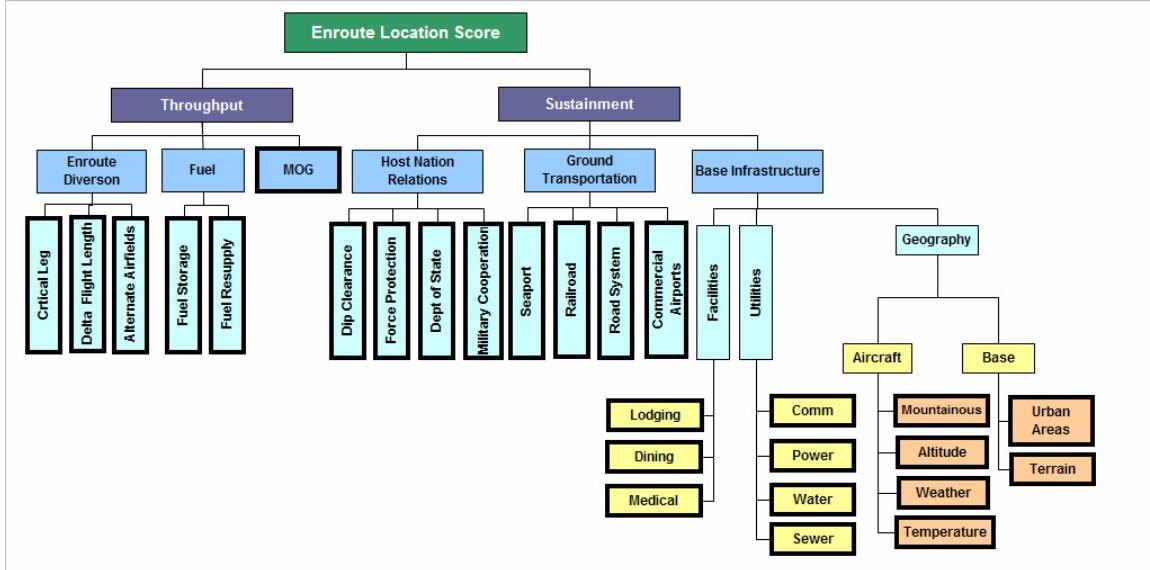


Figure 41: Tactical Submodel Hierarchy

Enroute Location Score		1.00	Throughput			1.00	Sustainment			1.00
		Throughput	Sustainment	Enroute Diversion	Fuel	MOG	Host Nation Relations	Ground Transportation	Base Infrastructure	
Local Weights	0.250	0.750	0.571	0.143	0.143	0.286	0.445	0.110	0.445	
Global Weights	0.250	0.750	0.143	0.036	0.036	0.072	0.334	0.083	0.334	
Enroute Diversion			Host Nation Relations			1.00			1.00	
Critical Leg	0.333	0.500	0.167	0.250	0.750	0.112	0.333	0.222	0.333	
Global Weights	0.048	0.071	0.024	0.009	0.027	0.037	0.111	0.074	0.111	
Ground Transportation			Base Infrastructure			1.00			1.00	
Seaport	0.105	0.105	0.263	0.527	0.154	0.231	0.615	0.667	0.333	
Railroad	0.009	0.009	0.022	0.043	0.051	0.077	0.205	0.137	0.068	
Road System	0.009	0.009	0.022	0.043	0.051	0.077	0.205	0.137	0.068	
Facilities			Utilities			1.00			1.00	
Lodging	0.625	0.313	0.062	0.067	0.400	0.267	0.266			
Dining	0.032	0.016	0.003	0.005	0.031	0.021	0.021			
Medical	0.032	0.016	0.003	0.005	0.031	0.021	0.021			
Aircraft			Base			1.00			1.00	
Mountainous	0.250	0.125	0.500	0.125	0.833	0.167				
Altitude	0.034	0.017	0.068	0.017	0.057	0.011				
Weather	0.034	0.017	0.068	0.017	0.057	0.011				
Temperature	0.034	0.017	0.068	0.017	0.057	0.011				

Figure 42: Tactical Submodel Weighting

Operational Model Value Functions

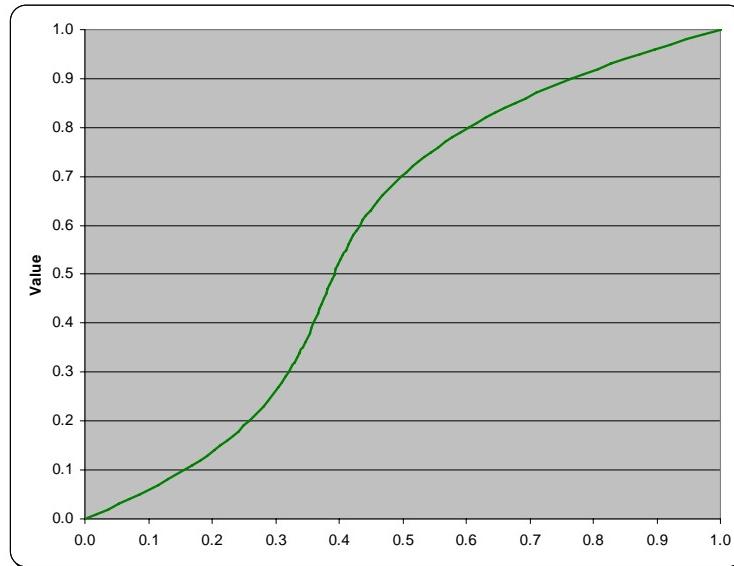


Figure 43: Probability of Utilization SDVF

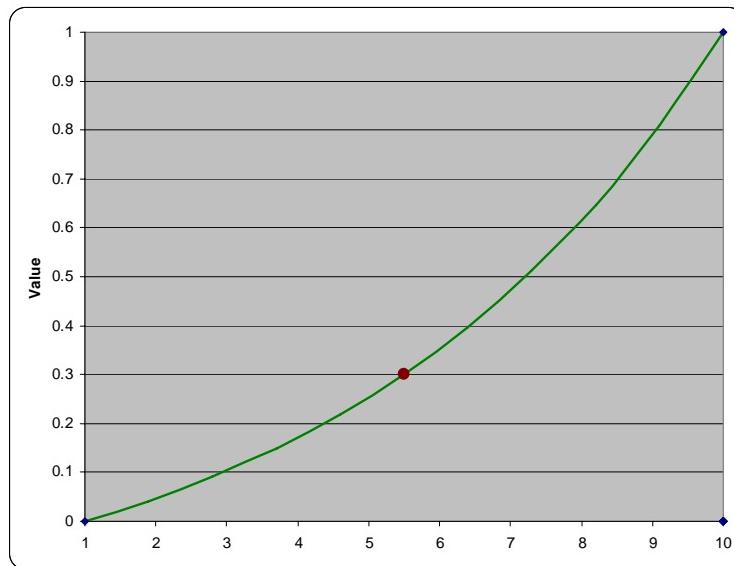


Figure 44: National Security SDVF

Operational Model Weighting

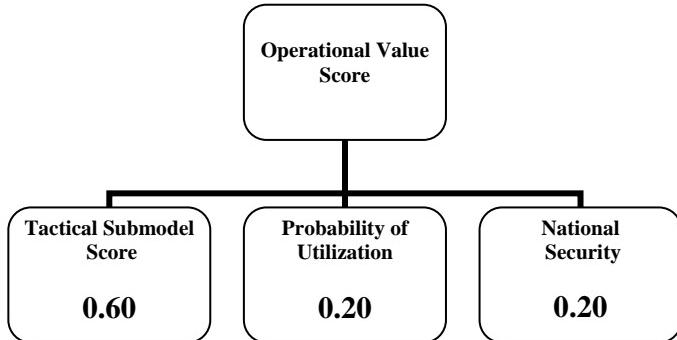


Figure 45: Operational Model Weighting

National Security Measure Definitions

Table 18: National Security Measure Definitions

NATIONAL SECURITY IMPORT		Minor Allies	Major Allies	US
1	None. Region has no impact in any area for US or allies.			
2	Slight. Instability/crisis in region has minimal impact on some US minor allies, but not the US or its major allies.			
3	Minimal. Instability/crisis in region has major impact on some minor allies or minimal impact on some major allies. No impact on the US.			
4	Low. Instability/crisis in region has significant impact on some minor allies and minimal impact on some major allies. No impact on the US.			
5	Low/Moderate. Instability/crisis in region has significant impact on some minor allies. Minimal impact on major allies and the US.			
6	Moderate. Instability/Crisis has minimal impact on major allies and minimal impact on US.			
7	Moderate/High. Instability/Crisis has major impact on major allies and minimal impact on US.			
8	High. Instability/Crisis has major impact on major allies and major impact on US.			
9	Very High. Instability/Crisis has significant impact on major allies and major impact on US.			
10	Extremely High. Instability/Crisis definitely has significant impact on US and major allies.			

Appendix 2. GERBIL User's Guide

The following pages contain the user's guide for GERBIL Version 1.0.



Global En Route Basing Infrastructure Location (GERBIL) Model

Version 1.0

User's Guide



May 2006

Global En Route Basing Infrastructure Location (GERBIL) Model User's Guide

Version 1.0

May 2006

Maj Alex Miravite
Maj Charles Schlegel
Air Force Institute of Technology
Department of Operational Sciences
2950 Hobson Way
Wright-Patterson AFB, OH 45433

TABLE OF CONTENTS

About GERBIL	58
Model Development.....	58
Tactical Submodel	58
Operational Value Hierarchy	59
Software Notes.....	59
Enabling Macros	59
Enabling ActiveX Controls.....	60
Closing Error	60
Application Interface	60
Saving GERBIL Output.....	63
Saving/Restoring Model	63
Integrated User Help.....	64
Origin/Destination Airfields.....	64
Latitude/Longitude format.....	64
Destination Attributes	65
Probability of Utilization	65
National Security	65
Adding Uncertainty.....	66
Throughput	66
En Route Alternatives Module	66
Measure Definitions.....	67
Airfield Input Form.....	67
Latitude/Longitude Format	67
Continuous Measures.....	68
Categorical Measures.....	68
Assigning Probability	68
Distance calculations	69
Critical Leg	69
Delta Flight Length.....	69
Within the Lens Determination.....	70
View Tactical Submodel Module.....	71
Measure Definitions.....	71
Value Functions	71
Continuous Increasing	72
Continuous Decreasing	73
Categorical	73
Save/Restore Value Functions	74
Weighting.....	75
Local vs Global	75
Weighting by Branch	75
Swing Weighting	75
Showing All Weights.....	76
Global Weights Chart	77
Sorted Global Weights Chart	78
Save/Restore Weights	78

Running the Tactical Submodel	79
Value Scores	79
Rankings Chart.....	80
Displaying by different Tiers	81
Copying Value/Rank Chart Results to File.....	81
Sensitivity Analysis	81
Local Proportional	81
Global Proportional.....	82
Copying Sensitivity Analysis Results to File	82
Running an Operational Scenario.....	84
Origin Selection	85
Operational Value Hierarchy	85
Weighting the Operational Value Hierarchy.....	85
Swing Weighting	85
Value Functions	85
Probability of Utilization and the Prospect Function.....	85
National Security Index Function	86
Operational Scenario Results.....	86
Scenario Scores.....	86
Scenario Top 10 Chart	87
Copying Scores/Chart Results to File	87
Scenario Sensitivity Analysis	88
Copying Sensitivity Analysis Results to File.....	88

LIST OF FIGURES

Figure 46: Tactical Submodel Diagram	58
Figure 47: Operational Value Hierarchy Diagram.....	59
Figure 48: Enabling Macros.....	59
Figure 49: Enabling ActiveX Controls Dialog	60
Figure 50: Closing error message	60
Figure 51: GERBIL Main Menu.....	61
Figure 52: Save/Restore Model Dialog.....	63
Figure 53: Origin/Destination Airfield Spreadsheet	64
Figure 54: Add Destination Airfield Dialog	65
Figure 55: En route Alternatives Module Spreadsheet	66
Figure 56: Airfield Input Form Dialog	67
Figure 57: Triangle Distribution Dialog	68
Figure 58: Empirical Distribution Dialog	69
Figure 59: View Tactical Submodel Module.....	71
Figure 60: Continuous Increasing SDVF.....	72
Figure 61: Continuous Decreasing SDVF	73
Figure 62: Categorical SDVF	74
Figure 63: Save/Restore Value Function Dialog	74
Figure 64: Sample Local Weighting by Branch	75
Figure 65: Show All Weights Spreadsheet	77
Figure 66: Global Weights Chart.....	78
Figure 67: Sorted Global Weights Chart	78
Figure 68: Save/Restore Weights.....	78
Figure 69: Run Tactical Submodel Menu.....	79
Figure 70: Value Scores Spreadsheet	80
Figure 71: Tactical Score Rankings Chart.....	80
Figure 72: Sample Local Proportional Sensitivity Analysis.....	82
Figure 73: Sample Global Proportional Sensitivity Analysis	82
Figure 74: Operational Scenario Menu.....	84
Figure 75: Prospect Function for Probability of Utilization	86
Figure 76: Scenario Score Matrix	87
Figure 77: Scenario Rankings Chart	87
Figure 78: Sample Scenario Sensitivity Analysis Chart.....	88

About GERBIL

Model Development

The Global En Route Basing Infrastructure Location (GERBIL) Model was developed for a line of research sponsored by US Transportation Command (TRANSCOM).

The model was developed as a graduate research project by Air Force Institute of Technology (AFIT) Operational Analysis students, Maj Alex Miravite and Maj Charles Schlegel with academic advisor Lt Col Raymond Staats. The GERBIL was adapted from the thesis studies of AFIT graduates Capt Michael Sere (2005) and Capt Marykathryn Tharaldson (2006), who began the application of the decision analysis technique of value-focused thinking to TRANSCOM's global en route basing problem.

For detailed information on the development and utility of the GERBIL Model, see Miravite and Schlegel (2006) and Tharaldson (2006).

Tactical Submodel

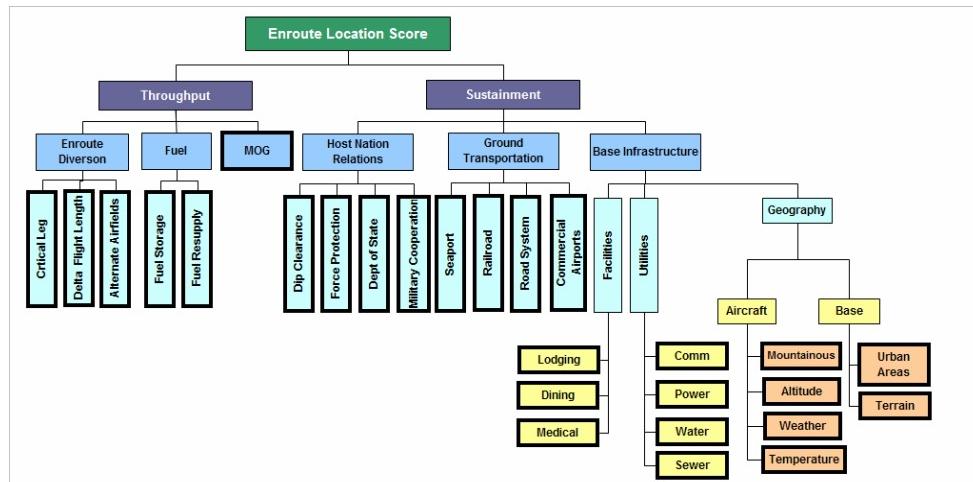


Figure 46: Tactical Submodel Diagram

The Tactical Submodel is used to evaluate all en route alternatives against a single origin-destination pair. The Tactical Submodel value hierarchy is adapted from the March 2006 AFIT Thesis of Capt Marykathryn Tharaldson. It is a 6-tiered value-focused hierarchy that measures the score of an en route location given an origin and destination pairing.

The en route location value score is determined by the evaluating the weighted values of 27 individual measures.

Operational Value Hierarchy

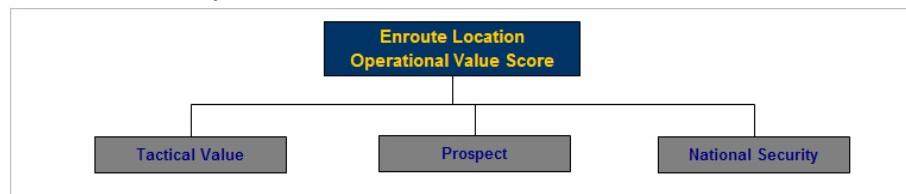


Figure 47: Operational Value Hierarchy Diagram

The operational value hierarchy evaluates the use of en route alternatives when considering a single specified origin and its reach towards all possible destinations.

Given an origin, the scenario is run to evaluate all destinations. Each en route alternative is evaluated using the weighted values of the three measures in Figure 47 which make up the individual measures scores: tactical value from the Tactical Submodel, the Prospect value from the probability of utilizing the destination airfield to access the destination region, and the national security value from the level of national security interest associated with the destination region. This process is repeated for the en route alternative for each accessible destination. The values for all accessible destinations for an alternative are then summed to arrive at an overall value score for that en route alternative.

Software Notes

The GERBIL User-Interface was developed using Microsoft Excel 2003 Visual Basic for Applications (VBA) Version 6.3. Programming code for the editing of single-dimensional value functions was adapted from code developed by Lt Col Jeffrey Weir, Department Head, AFIT Operational Sciences.

Full GERBIL functionality requires the use of both macros and ActiveX controls.

Enabling Macros

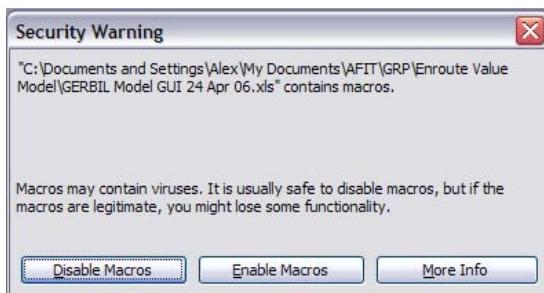


Figure 48: Enabling Macros

To enable macros, simply choose Enable Macros when prompted by Excel (Figure 48). If this dialog box does not appear, you may be required to change the Excel security settings. To modify the Excel security settings:

- | | |
|--------|---|
| Step 1 | Select <u>Tools</u> from the Excel pull down menu |
| Step 2 | Select <u>Options...</u> |
| Step 3 | Go to the <u>Security Tab</u> |
| Step 4 | Click on the <u>Macro Security...</u> button |

- | | |
|--------|-------------------------------------|
| Step 5 | Choose <u>Medium Security Level</u> |
| Step 6 | Click OK twice |
| Step 7 | Close Excel and re-open. |

Enabling ActiveX Controls

What is an ActiveX control?

“ActiveX control: A control, such as a check box or button that offers options to users or runs macros or scripts that automate a task. You can write macros for the control in Microsoft Visual Basic for Applications or scripts in Microsoft Script Editor.”, *Microsoft Excel Help*

GERBIL uses one ActiveX control in its user-interface. The control used is an embedded spreadsheet in the dialog user-form, used to get an empirical probability distribution from the user in the VIEW ENROUTE ALTERNATIVES module, “Add Alternative” function.

The dialog to enable ActiveX controls (Figure 49) may be displayed immediately when the workbook is opened or not until it is required in the Add Alternative user-form.

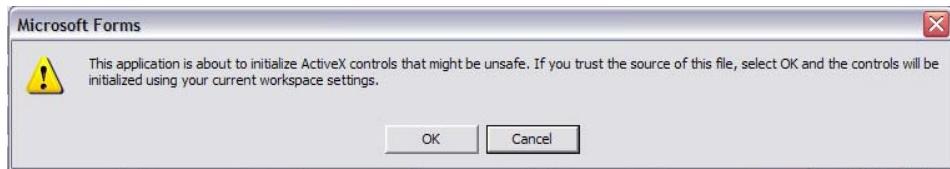


Figure 49: Enabling ActiveX Controls Dialog

Select **OK** to allow the ActiveX control required to input uncertain categorical alternative measures.

Closing Error

GERBIL uses multiple worksheets for different purposes. When required by the user-interface, all required links and references are established. Upon closing, the programmers have attempted to remove all hanging links/references. Upon occasion, you may receive the error message shown below when closing GERBIL. This does not affect future runs, select **OK**.

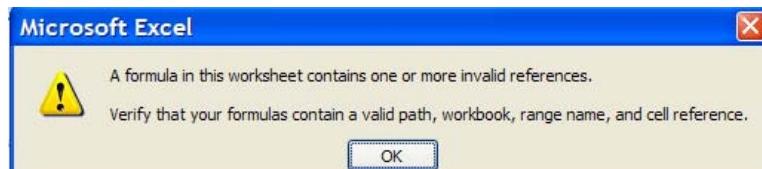


Figure 50: Closing error message

Application Interface

GERBIL is an Excel workbook protected to maintain the integrity of the mathematical model underlying the user-interface. VBA code is used to create a program application interface. The entire workbook is protected. The Excel formula bar is hidden. Vertical and horizontal scroll bars are only visible on sheets that require scrolling. The user navigates through the model by using command buttons linked to macros and manipulating data through user-form dialog boxes. Only unlocked cells can be accessed on worksheets themselves.

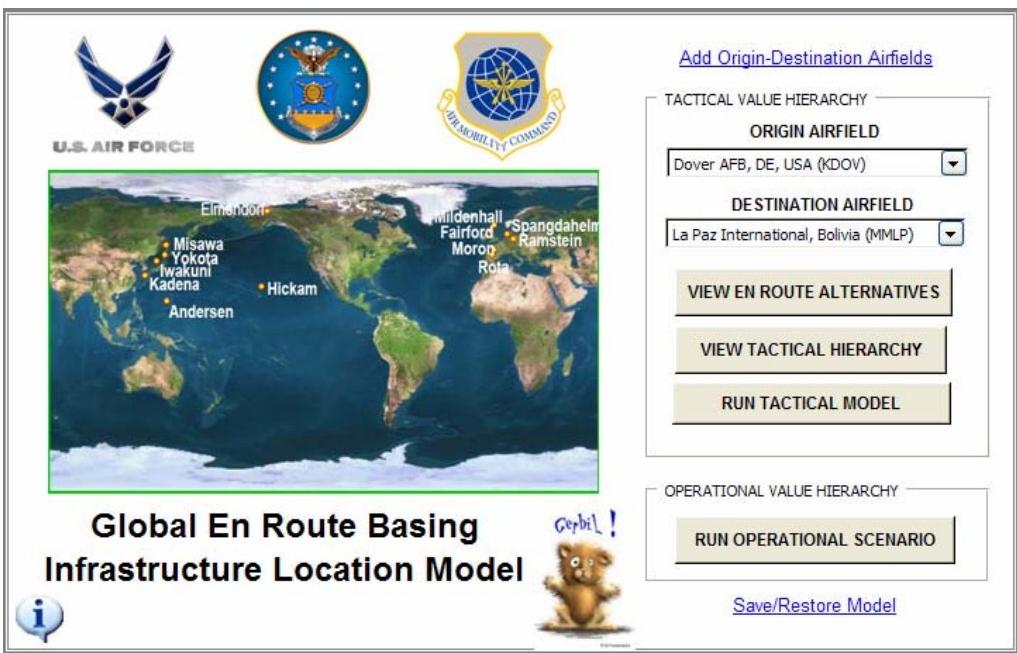


Figure 51: GERBIL Main Menu

Figure 51 shows the main menu interface for GERBIL. This is the screen which appears when the program is started. The program consists of two models, the Tactical Submodel and the Operational Value Hierarchy Model, and four modules. The En Route Alternatives Module is utilized by both the Tactical Submodel and the Operational Value Hierarchy Model.

Origin and destination airfields are edited by clicking on the [Add Origin-Destination Airfields](#) link. This edit function is detailed in the ORIGIN/DESTINATION AIRFIELDS section.

The ORIGIN AIRFIELD and DESTINATION AIRFIELD list boxes select the airfields that are used in running the Tactical Submodel. GERBIL uses the airfields selected from these list boxes to compute the great circle distances, in nautical miles, between the origin airfield and en route airfield alternatives and between the en route airfield alternatives and the destination airfield. NOTE: Depending upon where the list boxes were when last saved, it might be necessary to use the scroll up arrow to view additional airfields.

VIEW EN ROUTE ALTERNATIVES. Selecting this command button sends you to the En Route Alternatives Module. In this module you can view, add, edit, or delete en route alternatives and their corresponding 27 measures that are used in determining their tactical value score. See EN ROUTE ALTERNATIVES MODULE section for more information.

VIEW TACTICAL SUBMODEL. This command button sends you to the Tactical Submodel Hierarchy Module which allows you to view and edit the hierarchy used to evaluate en route alternatives for the origin-destination pair specified by the ORIGIN AIRFIELD and DESTINATION AIRFIELD list boxes. In this module, the user can view and modify the weighting associated with each hierarchy objective as well as view and modify the single-dimensional value functions

attributed to each measure. See **VIEW TACTICAL SUBMODEL MODULE** section for more information.

RUN TACTICAL SUBMODEL. This module is used to evaluate all en route alternatives for the given origin-destination pair specified by the ORIGIN AIRFIELD and DESTINATION AIRFIELD list boxes, by using the Tactical Submodel.

The top 10 alternatives for the given origin/destination pair are determined using the Tactical Submodel. A graphical bar chart can be displayed showing the top 10 results and is broken down by each measured objective, or by the tier of higher-level objectives above it. Additionally, the spreadsheet for each en route alternative's scores can be viewed by each measure. This spreadsheet also contains the scores for all the en route alternatives, not just the top 10.

Sensitivity analysis can also be performed on the value weighting for each measure. This analysis aids in evaluating the robustness of the top scoring alternatives.

See **RUNNING THE TACTICAL SUBMODEL** section for more information.

RUN OPERATIONAL SCENARIO. This module is used to run the Operational Value Hierarchy. Here, the model determines the en route alternatives that are valued the most given a specified origin airfield. In this module, the user specifies the origin airfield of interest. For each destination, each alternative is evaluated according to its Tactical Submodel score, its probability of use to access the destination region, and the level of national security interest for the destination region. The module sums up these values for an alternative for each destination the alternative can access.

The top 10 en route alternatives for the given origin airfield can be displayed in a rankings bar chart. Each en route alternative score can also be viewed in spreadsheet format. This spreadsheet also contains the scores for all the en route alternatives, not just the top 10.

The single-dimensional value functions for both the probability of using a specific en route alternative to access the destination region and the level of national security interest for the destination region can be viewed. Only the single-dimensional value function for the level of national security interest for the destination region can be modified. There is no single-dimensional value function for the Tactical Submodel since it is a direct input from a lower – level hierarchy, therefore it can not be viewed or modified from this module. For modifications to the Tactical Submodel see **RUN TACTICAL SUBMODEL** above. The weightings of the three measures for the Operational Value Hierarchy can be viewed and modified.

Sensitivity Analysis on the weightings used in the Operational Value Hierarchy can be performed to determine the robustness of scenario results.

See **RUNNING AN OPERATIONAL SCENARIO** section for more information.

Saving GERBIL Output

When desired, GERBIL can output model results to another Excel file. This allows the user to store results while being able to format and copy charts for results presentation.

Results worksheets are written to GERBIL MODEL RESULTS.XLS workbook in the user's MyDocuments folder. All macros are stripped from the copied sheets, so the results file should not require the use of macros. The results file worksheets are independent from the main GERBIL application workbook, containing no links back to the main application.

The following GERBIL outputs can be saved to the results file:

- Tactical Submodel scores with top 10 rankings chart
- Tactical Submodel sensitivity analysis charts
- Operational value scores with top 10 rankings chart
- Operational values hierarchy sensitivity analysis charts

Saving/Restoring Model

The Save/Restore dialog box (Figure 52) can be accessed by clicking on [Save/Restore Model](#) from the main menu. Select the desired option and press Update to execute the option. The dialog box will not close until Close is selected.

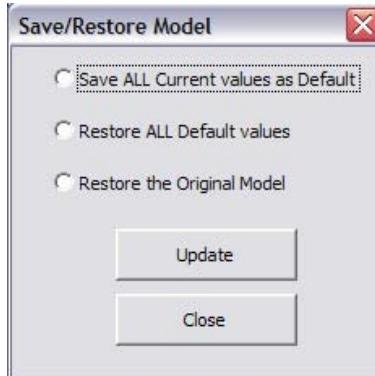


Figure 52: Save/Restore Model Dialog

Save ALL Current values as Default. Saves the current weights and value functions for all Tactical Submodel measures and for the Operational Value Hierarchy measure for National Security. The saved values will be saved as default and can be restored using the Restore ALL Default values option.

Restore ALL Default values. Restores the weights and value functions saved as default. The initial default values are the same as the original model values. Default values can be changed and saved as the user customizes the model.

Restore the Original Model. Restores the original Tactical Submodel weightings and the original single-dimensional value functions for the Tactical Submodel and for the Operational Value Hierarchy measure for National Security as provided by AFIT.

(Note: Default or Original weights are not maintained for the Operational Value Hierarchy.
Only the most recent valid weights are saved.)

Integrated User Help



Help is available throughout the program and can be found by clicking on any "i" icon found on a worksheet or by clicking on Help in a user-form dialog box.

Additionally, many user-forms contain ControlTipText help. Hold the mouse cursor over a data field and a help box will appear displaying help, range information, or suggested/required formatting.

Origin/Destination Airfields

This area of GERBIL is used to specify the collection of origin and destination airfields that are used for running the models. In this program area, the user can add/delete origins and/or add/edit/delete destination airfields.

ORIGIN		Return to Main		Decimal Degree Notation (Automatically Calculated)		Add Destination		Click for table of National Security measure definitions		Probability of Utilization		National Security		Throughput Capability		Decimal Degree Notation (Automatically Calculated)				
N/S	Deg MM.MM	E/W	Deg MM.MM	Latitude	Longitude	Edit Destination		Delete		Min	Mode	Max	Latitude	Longitude	N/S	Deg MM.MM	E/W	Deg MM.MM	Latitude	Longitude
Dover AFB, DE, USA (KDOV)	N 39 7 77.0	W 75 27.950	N 39 13.0	-75.466						0.45	5	5	5	S 22 28.79	E 17 26.26	S 22 28.79	E 17 26.26	N 22 28.79	17.47	
Fairford AB, UK (EGVA)	N 51 40 99	W 0 47.40	N 51 40 00	-0.799						0.10	7	5	10	S 33 57.90	E 1 26.00	S 33 57.90	E 1 26.00	N 33 57.90	19.60	
Charleston, SC, USA (KCHS)	N 32 53.92	W 80 2.43	N 32 53.92	-80.041						0.40	7	3	6	S 11 35.47	E 27 31.95	S 11 35.47	E 27 31.95	N 11 35.47	27.53	
Middleton, England (EGUN)	N 52 21.72	E 0 29.18	N 52 21.72	0.496						0.30	5	5	5	S 15 19.95	E 28 27.16	S 15 19.95	E 28 27.16	N 15 19.95	20.45	
Morón AB, SP (LEMO)	N 37 10.50	E 5 37.00	N 37 10.50	5.617						0.50	5	5	5	S 6 52.69	E 39 12.16	S 6 52.69	E 39 12.16	N 6 52.69	39.20	
Rota NS, SP (LERT)	N 36 38.70	E 6 21.00	N 36 38.70	6.360						0.10	5	5	5	N 19 5.32	E 72 52.07	N 19 5.32	E 72 52.07	N 19 5.32	72.87	
Spangdahlem AB (ETAD)	N 49 58.40	E 6 41.50	N 49 58.40	6.692						0.60	7	7	7	S 7 18.78	E 72 24.65	S 7 18.78	E 72 24.65	N 7 18.78	7.31	
Ramstein AB, GE (ETAR)	N 49 26.20	E 7 36.00	N 49 26.20	7.600						0.10	5	5	5	N 13 4.27	E 79 41.47	N 13 4.27	E 79 41.47	N 13 4.27	72.41	
										0.05	5	5	5	N 7 10.87	E 79 63.07	N 7 10.87	E 79 63.07	N 7 10.87	7.18	
										0.05	2	2	2	N 27 41.78	E 85 21.55	N 27 41.78	E 85 21.55	N 27 41.78	85.36	

Figure 53: Origin/Destination Airfield Spreadsheet

Latitude/Longitude format

Both origin and destination airfields require latitude and longitude in the degrees-minutes-decimal (DMD) format.

The minutes fields must be entered in minutes-decimal format and not in minutes-seconds.

example N 75 deg 25.55 mm.mm

If data is provided in degrees-minutes-seconds format, convert to degrees-minutes-decimal by dividing the seconds by 60 and appending that decimal value to the integer minutes value.

example Degrees-Minutes-Seconds (DMS) format
N 75 deg 25'33"

Dividing 33 seconds by 60 gives 0.55.

Degrees-Minutes-Decimal (DMD) format
N 75 deg 25.55 mm.mm

The DMD latitude/longitude information entered in the dialog boxes are further transformed to Degree-Decimal format by GERBIL. Southern latitudes and western longitudes are converted automatically by GERBIL to negative values for use in great circle computations.

Destination Attributes

In addition to latitude and longitude information, each destination airfield is given attributes that are used in the operational scenario run. These attributes are the Probability of Utilization and National Security Interest. There is future expansion to include the Throughput Capability of the destination as an attribute. The National Security attribute can be entered stochastically. If the user is unsure of what value to assess for National Security, the user can choose to enter the information with uncertainty.

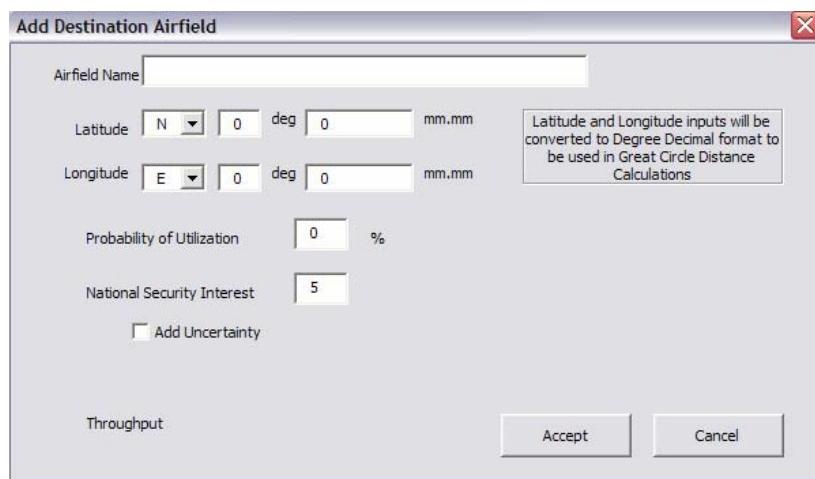


Figure 54: Add Destination Airfield Dialog

Probability of Utilization

Probability of Utilization measures how likely it is that a crisis or situation would arise in the destination region requiring the given destination to be used as an Aerial Port of Debarkation (APOD). Examples are humanitarian crises, noncombatant evacuation orders, show of force, and military action. It is important to note that the presence of another APOD in a region could decrease the probability of a destination being utilized. A value from 0-100 percent is entered.

National Security

National Security Index measures the importance of the region which would be served by the destination airfield. For example, the Mid East region around Kuwait would score very high due to the importance of its oil fields to the world economy. A place like Sudan might not score as high due to its lower level impact on the National Security of the United States in a number of areas. A value from 1 to 10 is entered.

Click on [Click for table of National Security measure definitions](#) for a graphical explanation of the values from 1 to 10 for the National Security measure.

Adding Uncertainty

If the user is unsure what value to assess to National Security, the user can enter a minimum, most likely, and maximum value for National Security. This is a stochastic triangle distribution assessment which allows the user to input a spread of values. This uncertainty is assessed by GERBIL for a risk neutral decision maker. Future improvements to GERBIL may allow for an interview process of the user to determine the degree of risk tolerance (risk-averse or risk-seeking).

Throughput

This potential future attribute for destinations measures the cargo and passenger throughput capability of the destination airfield.

En Route Alternatives Module

Figure 55: En route Alternatives Module Spreadsheet

This module is used for storing the data for the en route airfield alternatives. In this module, you can add/edit/delete alternatives information. The user inputs an airfield's latitude and longitude information and the values for the measures used to compute the Tactical Submodel score.

Airfields are sorted according to whether or not they are in the lens, then alphabetically in the two subcategories. The order of en route alternatives varies for different origin-destination pairs.

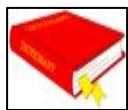
Distance computations are based on the selected ORIGIN AIRFIELD and DESTINATION AIRFIELD and are computed to determine the Critical Leg and Delta Flight Length measures. Distance computations are also used to determine if an en route alternative is “within the lens”.

Alternatives names are formatted in RED for the following conditions:

- Critical Leg exceeds the maximum critical leg distance defined in the Tactical Submodel single-dimensional value function (SDVF) for Critical Leg
 - Dip Clearance is equal to “None”
 - Dept of State is equal to “Unacceptable”

“?” indicates that a measure was input with uncertainty and assigned a probability distribution in the Airfield Input form. To view the assigned probability for a measure, the user must return to the Airfield Input form by selecting Edit Alternative. See Airfield Input Form for more details.

Measure Definitions



Click on this icon to view the detailed definitions of all 27 Tactical Submodel measures and the Operational Value Hierarchy measures for Probability of Utilization and National Security Import.

Airfield Input Form

The Airfield Input Form is used for adding or editing an alternative.

ControlTipText help is available on the form. Place the mouse over the input box and help will be displayed.

There are input boxes for the Airfield Name, Latitude/Longitude information, and 25 of the 27 measures used to determine the Tactical Submodel Score. Each measure can be assigned a probability if the measure is not known with certainty. Click on the “?” command button to view or edit the probability, which is not actually assigned until the Assign Probability checkbox is checked.

Add a new airfield alternative

Airfield Name: [Input Box]

Latitude: N 0 deg 0.00 mm.mm Longitude: E 0 deg 0.00 mm.mm

Latitude and Longitude inputs will be converted to Degree Decimal format to be used in Great Circle Distance Calculations

Assign Probability?		View/Edit Probability	
MOG	0	<input type="checkbox"/>	?
Critical Leg	Values automatically calculated on Alternatives page	<input type="checkbox"/>	?
Delta Flight Length	0	<input type="checkbox"/>	?
Alt Airfields	0	<input type="checkbox"/>	?
Fuel Storage	0	<input type="checkbox"/>	?
Fuel Resupply	0	<input type="checkbox"/>	?
Diplomatic Clearance	None	<input type="checkbox"/>	?
Force Protection	Complete	<input type="checkbox"/>	?
Dept of State	Unacceptable	<input type="checkbox"/>	?
Military Cooperation	Poor	<input type="checkbox"/>	?
Seaport	0	<input type="checkbox"/>	?
Railroad	0	<input type="checkbox"/>	?
Road System	0	<input type="checkbox"/>	?
Commercial Airport	0	<input type="checkbox"/>	?
Lodging	Inadequate	<input type="checkbox"/>	?
Dining	Inadequate	<input type="checkbox"/>	?
Medical	No	<input type="checkbox"/>	?
Comm	None	<input type="checkbox"/>	?
Power	None	<input type="checkbox"/>	?
Water	None	<input type="checkbox"/>	?
Sewer	None	<input type="checkbox"/>	?
Mountainous	Yes	<input type="checkbox"/>	?
Altitude	Yes	<input type="checkbox"/>	?
Weather	Heavy	<input type="checkbox"/>	?
Temperature	Yes	<input type="checkbox"/>	?
Urban Areas	Yes	<input type="checkbox"/>	?
Terrain	Yes	<input type="checkbox"/>	?

Show Measure Definitions

Accept

Cancel

Place mouse over input box to view the range for the measure inputs

Figure 56: Airfield Input Form Dialog

Latitude/Longitude Format

For correct Latitude/Longitude format, see previous discussion in ORIGIN/DESTINATION AIRFIELDS, Latitude/Longitude format (page 64).

Continuous Measures

Continuous measures are those objectives that are measured numerically. This is identifiable by the measure having an input box that is not a pull-down list box. ControlTipText will list the range used by the measure's SDVF to assess its value score.

If the exact measure input is not known, the user can assign a triangular probability distribution that will assess a minimum, most likely, and maximum value for the measure.

Categorical Measures

Categorical measures are those objectives that are broken down into categories. This is identifiable by the measure having an input box that is a pull-down list box. ControlTipText is not available for this input as only inputs from the list box can be entered.

If the user cannot assign a categorical value with certainty, the user can assign an empirical probability distribution that will assess the probability for each possible categorical value.

Assigning Probability

GERBIL allows for the input of stochastic measures for the airfield alternatives. This allows a user to enter values using a probability distribution when the exact value for a measure is uncertain. GERBIL Version 1.0 assesses the risk with a risk neutral approach.

A user can assign a triangular distribution to objectives that are measured continuously. The user must specify the minimum, most likely, and maximum values for the objective.

The default values for triangular distributions are determined by the bounds of the measure's SDVF at the time the en route alternative is added. The minimum and maximum values coincide with the lower and upper bounds of the value function, and the default mode is equal to the midpoint between the lower and upper bounds.



Figure 57: Triangle Distribution Dialog

For categorical measures, the user can specify an empirical distribution assessing the probability for each categorical value.

(Note: The spreadsheet in the Empirical Distribution Dialog box is the only ActiveX control used in the program)

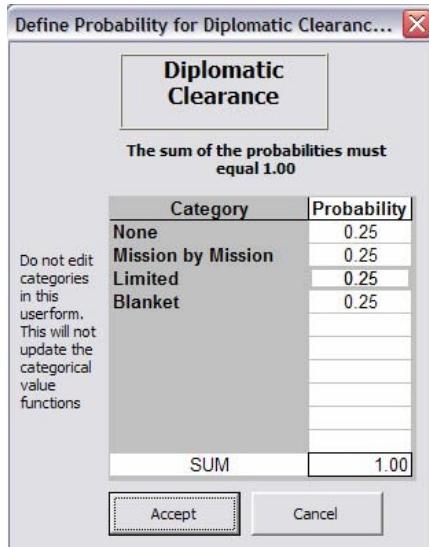


Figure 58: Empirical Distribution Dialog

The default probabilities are set such that each categorical value is equally likely.

When specifying an empirical distribution, the sum of the probabilities must equal one. Click in the cell you wish to modify. Press ENTER or the Down-Arrow to move to the next cell.

Do not modify the categories in the shaded area as this will have no affect on the actual categories. If you wish to modify the categories used for a measure, the SDVF must be modified in the View Tactical Submodel Module.

(Note: To assign a probability distribution, the check box next to the measure must be marked)

Distance calculations

GERBIL performs great circle distance calculations to determine the distance between the origin airfield and the en route alternative and between the en route alternative and the destination airfield.

Critical Leg

Critical Leg measures the longer leg of the two-leg flight, the distance from the origin to the en route or the distance from the en route to the destination. The maximum critical leg is specified in the Critical Leg SDVF in the View Tactical Submodel Module. Any alternative with a Critical Leg value greater than the maximum is formatted RED.

Delta Flight Length

Delta Flight Length is the difference between the length of the direct flight from the origin to destination and the length of the two-leg flight through the proposed en route base. A large delta penalizes the potential en route airfield because of the increased use of fuel and flight hours. The goal is to find an en route with a flight path as close to the original flight path as feasible.

If an alternative has a Critical Leg value greater than the maximum specified critical leg, the Delta Flight Length is given a value of 9999 to penalize it as a non-feasible alternative.

Within the Lens Determination

The Lens is a region specified by drawing two arcs, one from the origin and one from the destination, with a radius equal to the maximum planned range for a strategic airlift aircraft, currently 3500 nm. En route alternatives that fall within the lens are feasible alternatives for the given origin-destination pair.

The "Within the Lens?" column returns a TRUE value if the alternative is within a radius of the maximum Critical Leg distance from both the origin airfield and destination airfield.

View Tactical Submodel Module

The View Tactical Submodel Module allows the user to view and edit the value hierarchy used to determine the Tactical Submodel score for an en route alternative given a specific origin-destination pairing. In this module, the user can view and edit the SDVFs for the hierarchy's 27 measures. The user can also view and edit the weight that each objective imparts on the overall Tactical Submodel score.

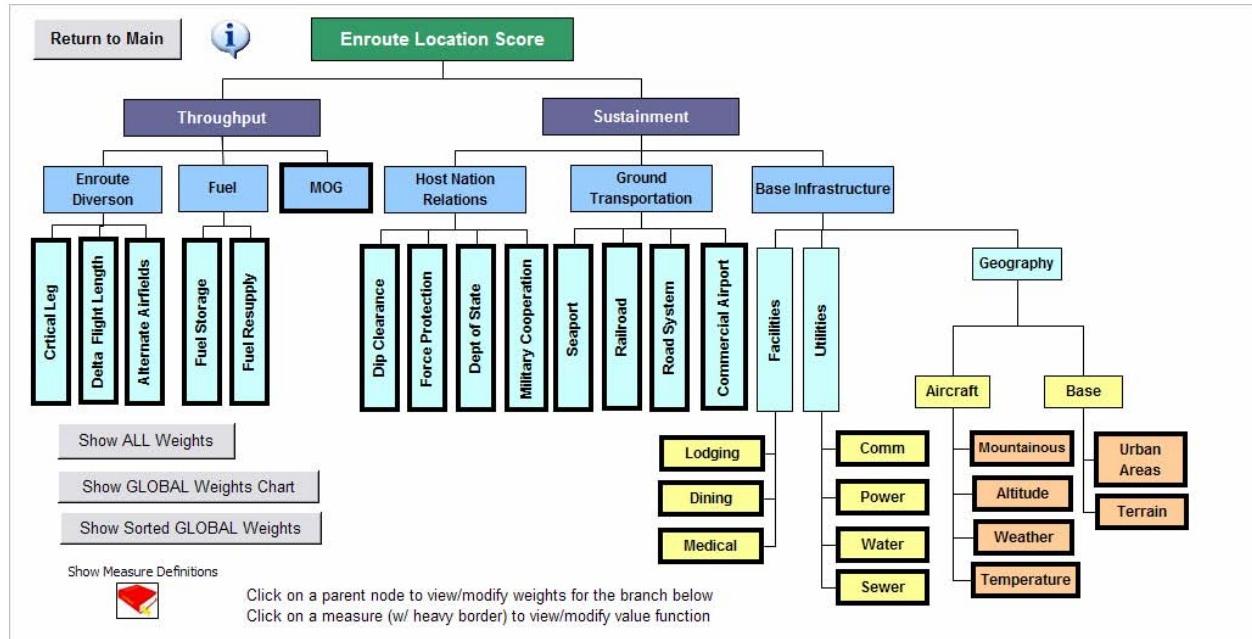


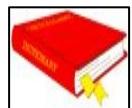
Figure 59: View Tactical Submodel Module

Parent Node	A higher-level objective with sub-objectives below it
Measure Node	An objective at the lowest level that is measured to determine its value and the value of the objectives above it. Identified by the heavy border.
Branch	A collection of objectives at the same level with the same Parent Node.
	The local weighting within a branch must sum to one.

To view the weighting of each branch, click on the parent node.

To view the SDVF for a measure, click on the measure node.

Measure Definitions



Click on this icon to view the detailed definitions of all 27 Tactical Submodel measures and the Operational Value Hierarchy measures for Probability of Utilization and National Security.

Value Functions

A single-dimensional value function (SDVF) returns the value of an objective given the measure of that objective. GERBIL uses continuous increasing, continuous decreasing, and categorical

SDVFs. The user can modify the SDVF according to their decision maker's assessment of the importance of the measure.

The ability to modify continuous and categorical SDVFs was adapted from code developed by Lt Col Jeffrey Weir, Department Head, AFIT Operational Sciences.

When modifying SDVFs, GERBIL allows access to unlocked cells on the worksheet.

If modifying a cell value, ensure you press ENTER before selecting a command button as the cell value update will not occur until ENTER is pressed.

Continuous Increasing

A continuous increasing SDVF is used for objectives that are measured with continuous measures and who's assessed value increases as the measure increases.

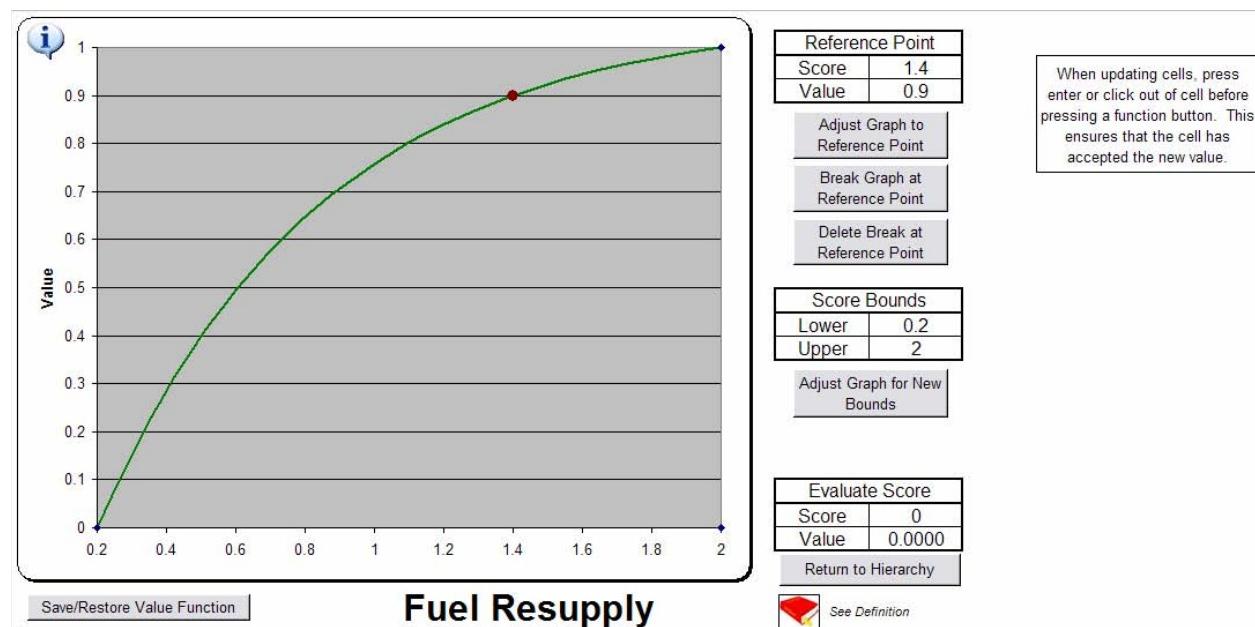


Figure 60: Continuous Increasing SDVF

This sheet allows the user to modify the form of the function to map the x-axis to a value (displayed on the y-axis) between zero and one. The function can be either exponential or piecewise linear. The default is a linear function.

By changing the values in the box labeled "Reference Point" the user can adjust the location of the "red" reference point on the graph. When the ADJUST GRAPH TO REFERENCE POINT button is pressed an exponential function will be drawn from the lower bound to the upper bound through the reference point.

To build piecewise linear functions or "S-curves", the user breaks the function at the reference point by pressing the BREAK GRAPH AT REFERENCE POINT button. Now the function on each side of the break point can be adjusted separately by moving the reference point to the appropriate section of the graph. This program is limited to 7 break points in the function.

To remove a break point simply put the coordinates into the reference point box and press the DELETE BREAK AT REFERENCE POINT button.

Finally, the user may change the upper and lower bounds by typing the new bounds into the Score bounds box and pressing the ADJUST GRAPH FOR NEW BOUNDS button. Remember to press ENTER after changing the cell values before pressing the ADJUST GRAPH FOR NEW BOUNDS button.

To find the value for a specific score, place the score into the Evaluate Score boxes and press ENTER to read the value below it.

Pressing the Return to Hierarchy button returns the user to the Tactical Submodel sheet.

Continuous Decreasing

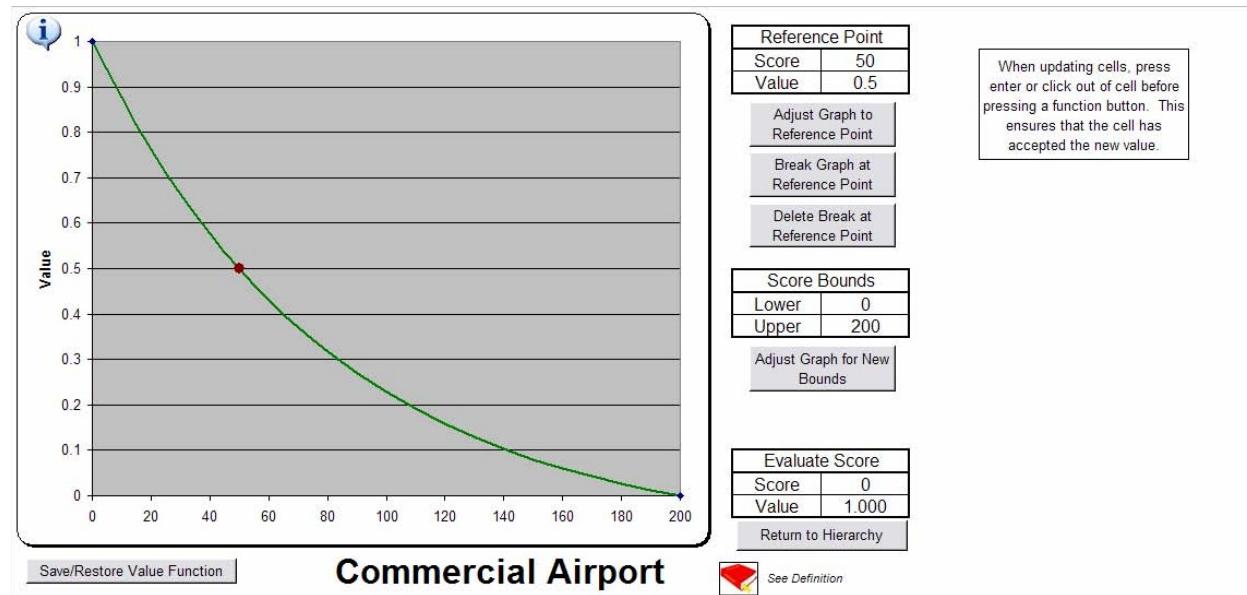


Figure 61: Continuous Decreasing SDVF

This sheet operates the same as the Continuous Increasing SDVF discussed above. The continuous decreasing SDVF differs in that the value decreases as the score for the measure increases.

Categorical

This type of measure can be used for qualitative or discrete measures. The user is limited to 10 categories. Simply type the name of the category into the appropriate column and provide its value.

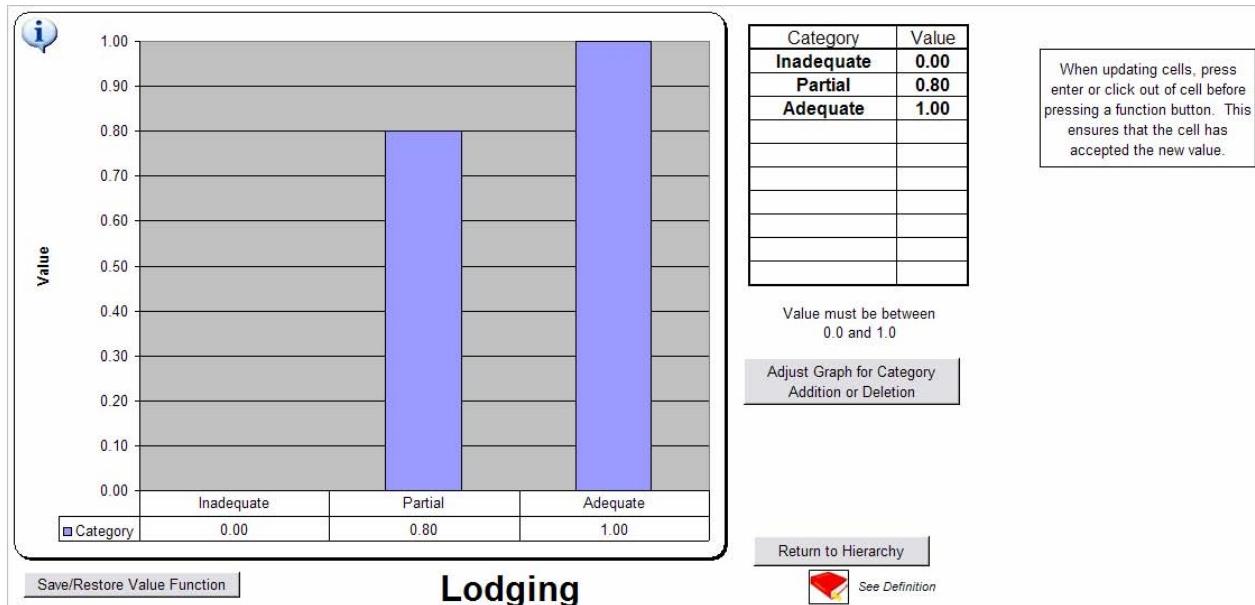


Figure 62: Categorical SPVF

Pressing the **ADJUST GRAPH FOR CATEGORY ADDITION OR DELETION** button will update the graph if the user adds or delete a category from the list. Press this button anytime a category or value is changed to update the categories in all modules of the program.

Save/Restore Value Functions

The Save/Restore dialog box (Figure 63) can be accessed by clicking on Save/Restore Value Function command button. Select the desired option and press Update to execute the option. The dialog box will not close until Close is selected.



Figure 63: Save/Restore Value Function Dialog

Save ALL Current VF as Default. Saves the current value function parameters. The saved values are saved as default and can be restored using the Restore Default VF option.

Restore Default VF. Restores the value functions saved as default. The initial default values are the same as the original model values. Default values can be changed and saved as the user customizes the model.

Restore Original VF. Restores the original value function provided by AFIT.

Weighting

The weights assessed to an objective determine what proportion of the overall score that objective contributes.

Local vs Global

Local Weights	Refers to the weighting of an objective in comparison to other objectives in its branch. The sum of weights in a branch must equal one.
Global Weights	Refers to the overall weight an objective has on the overall objective. The sum of global weights along the same level or tier of objectives must equal one.

GERBIL allows for modification of weights by modifying local weighting only. Modification of local weighting affects global weighting. Local weights can be modified by branch in the View Tactical Submodel screen or the user may choose to view all weights at once by selecting Show ALL Weights.

Weighting by Branch

Click on a Parent Node to view and edit the weights of the branch below it. The dialog box will allow a user to modify local weights while displaying the current global weights.

Values entered must be numerical. A technique is provided to aid the user in assessing local weights; this is the swing weighting technique discussed below.

In order to Accept New Weights, the sum of the weights must equal one. GERBIL allows flexibility of +/- .002 for rounding. Calculate Swing Weights will normalize any values so that they sum to one.



Figure 64: Sample Local Weighting by Branch

Swing Weighting

Two "swing" techniques are provided to aid the user in determine weighting for a branch of objectives.

(1) BY MOST IMPORTANT OBJECTIVE

Determine which objective you deem MOST important and assign that value a weight of 1.

For each of the remaining objectives, determine what proportion of importance each objective has with respect to the most important objective. Enter the decimal equivalent of that proportion.

Example with Objective 3 most important

	User Input	Swing Values
Objective 1	0.25	0.1073
Objective 2	0.33	0.1416
Objective 3	1	0.4292
Objective 4	0.75	0.3219

This swing weighting values Objective 3 four times as much as Objective 1, three times as much as Objective 2, and one-third more than Objective 4.

Click on CALCULATE SWING WEIGHTS to normalize the entered values so that they sum to 1.

(2) BY LEAST IMPORTANT OBJECTIVE

Determine which objective you deem LEAST important and assign that value a weight of 1.

For each of the remaining objectives, determine how much more important each objective is with respect to the least important objective.

Example with Objective 2 least important

	User Input	Swing Values
Objective 1	2.5	0.3333
Objective 2	1	0.1333
Objective 3	2.5	0.3333
Objective 4	1.5	0.2000

Click on CALCULATE SWING WEIGHTS to normalize the entered values so that they sum to 1.

Showing All Weights

Another method provided by GERBIL to modify weights is by clicking on the Show ALL Weights command button. This provides the user with a view of all branches at once. As the user modifies the local weights, the spreadsheet formulas automatically updated the affected global weights.

The sum of local weights for a branch must equal one. Error validating cells compute the sum next to the name of the Parent objective. If the sum of the branch equals one (+/- .002), the cells are formatted **GREEN**, otherwise the cells are formatted **RED**.

The sum of the global weights for the lowest-level objectives, also known as our measures, must also equal one. An error validating cell is included for this sum as well.

The user will not be able to leave this spreadsheet if any error validating cell is RED.

TOTAL GLOBAL WEIGHTS		1.00		Throughput		1.00		Sustainment		1.00		
		Throughput	Sustainment	Enroute Diversion	Fuel	MOG	Host Nation Relations	Ground Transportation	Base Infrastructure			
Local Weights	0.250	0.750	0.571	0.143	0.286	0.445	0.110	0.445				
Global Weights	0.250	0.750	0.143	0.036	0.072	0.334	0.083	0.334				
Enroute Diversion		1.00		Fuel		1.00		Host Nation Relations		1.00		
Critical Leg	Delta Flight Length	Alternate Airfields		Fuel Storage	Fuel Resupply	Dip Clearance	Force Protection	Department of State	Military Cooperation			
Local Weights	0.333	0.500	0.167	0.250	0.750	0.112	0.333	0.222	0.333			
Global Weights	0.048	0.071	0.024	0.009	0.027	0.037	0.111	0.074	0.111			
Ground Transportation			1.00		Base Infrastructure		1.00		Geography		1.00	
Seaports	Railroads	Road System	Commercial Airport		Facilities	Utilities	Geography		Aircraft	Base		
Local Weights	0.105	0.105	0.263	0.527	0.154	0.231	0.615		0.667	0.333		
Global Weights	0.009	0.009	0.022	0.043	0.051	0.077	0.205		0.137	0.068		
Facilities			1.00		Utilities		1.00					
Lodging	Dining	Medical		Communications	Power	Potable Water	Sewer					
Local Weights	0.625	0.313	0.062	0.067	0.400	0.267	0.266					
Global Weights	0.032	0.016	0.003	0.005	0.031	0.021	0.021					
Aircraft			1.00		Base		1.00					
Mountainous	Altitude	Weather	Temperature	Urban Areas	Terrain							
Local Weights	0.250	0.125	0.500	0.125	0.833	0.167						
Global Weights	0.034	0.017	0.068	0.017	0.057	0.011						

Figure 65: Show All Weights Spreadsheet

Global Weights Chart

Pressing Show GLOBAL Weights Chart displays a graphical representation of the global weights of each measure.

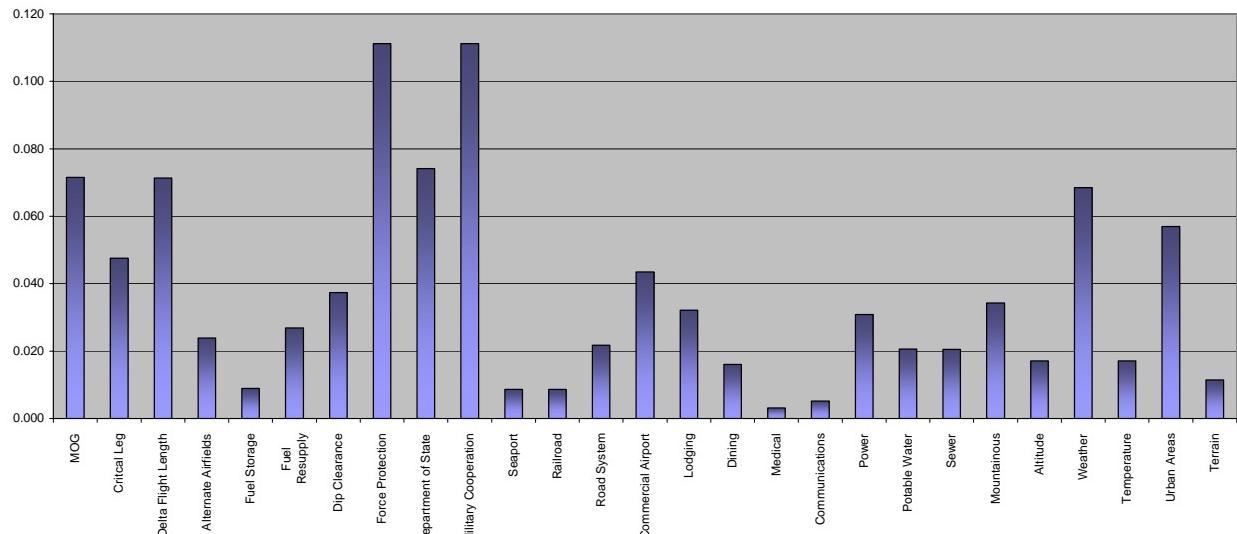


Figure 66: Global Weights Chart

Sorted Global Weights Chart

Pressing Show SORTED Global Weights Chart presents a graphical representation of the global weights, with the measures listed in order of importance from top to bottom.

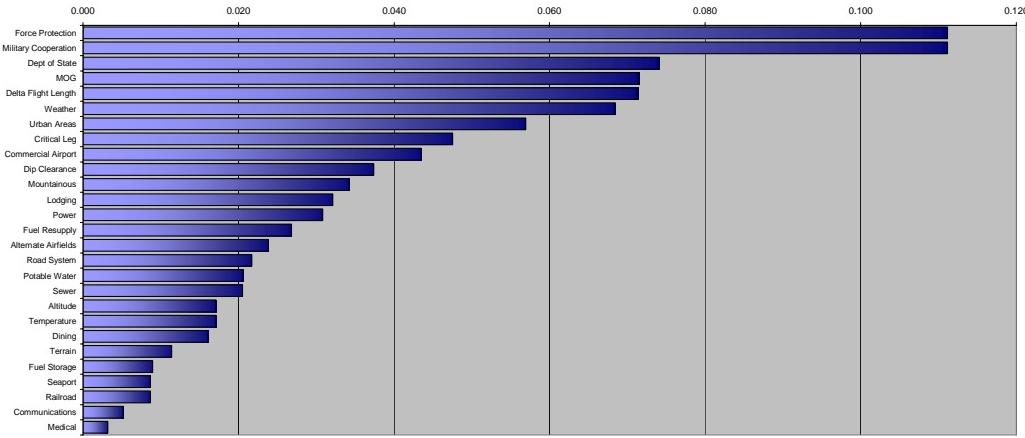


Figure 67: Sorted Global Weights Chart

Save/Restore Weights

The Save/Restore dialog box (Figure 68) can be accessed by clicking on Save/Restore Weights command button in the Show ALL Weights sheet. Select the desired option and press Update to execute the option. The dialog box will not close until Close is selected.

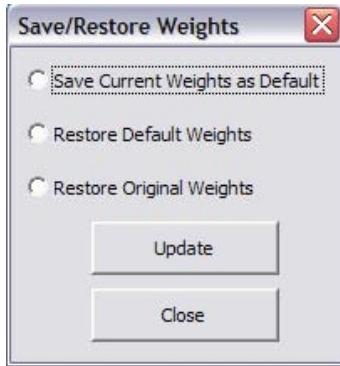


Figure 68: Save/Restore Weights

Save Current Weights as Default. Saves the current weights for the Tactical Submodel. The saved weights will be saved as default and can be restored using the Restore Default Weights option.

Restore Default Weights. Restores the weights saved as default. The initial default weights are the same as the original model values. Default weights can be changed and saved as the user customizes the model.

Restore Original Weights. Restores the original weights provided by AFIT.

Running the Tactical Submodel

This module is used to obtain GERBIL results for the Tactical Submodel. Using the ORIGIN AIRFIELD and DESTINATION AIRFIELD selected from the list boxes on the main menu, all en route alternative airfields are evaluated using the Tactical Submodel.

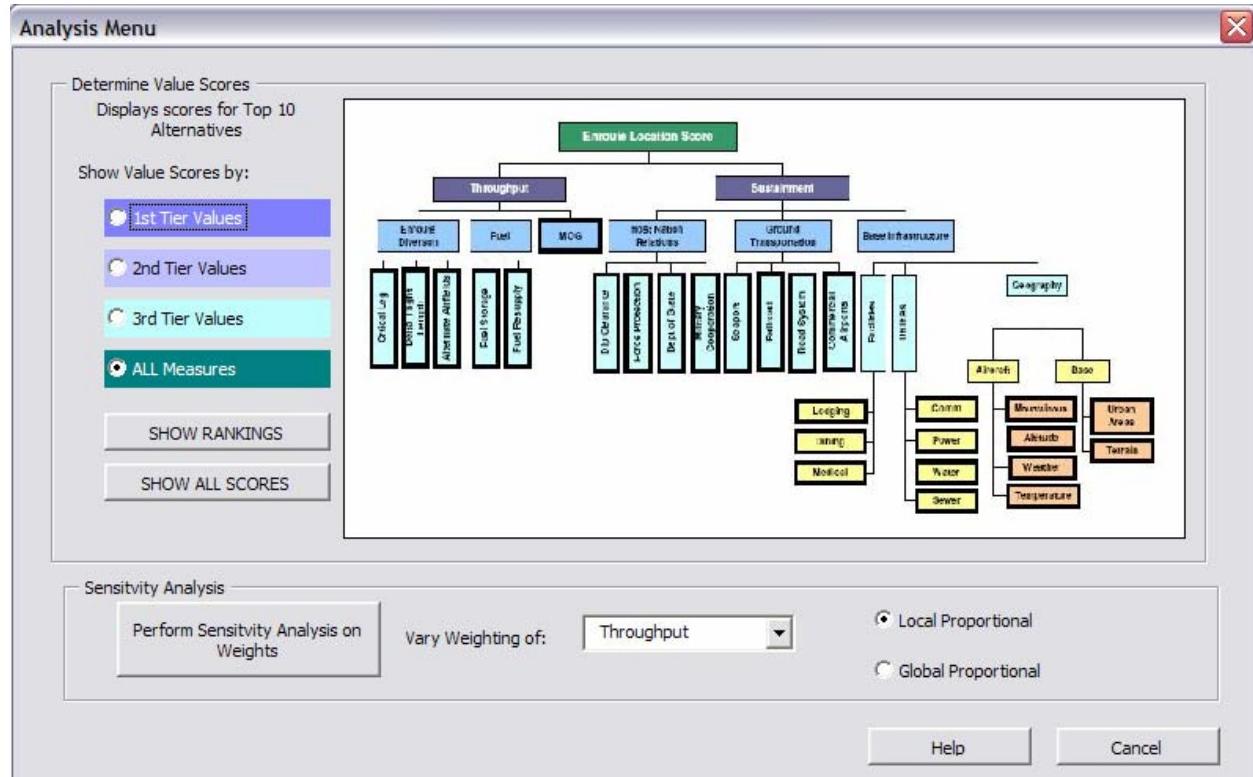


Figure 69: Run Tactical Submodel Menu

SHOW RANKINGS. Displays a chart of the top 10 en route alternatives for the origin-destination pairing selected. The chart can be broken down by all measures or by a tier of higher-level objectives.

SHOW ALL SCORES. Displays a spreadsheet showing the value scores for all en route alternatives broken down by all measures and by tier.

PERFORM SENSITIVITY ANALYSIS ON WEIGHTS. Displays a line chart depicting how the results would vary if the weighting for the selected objective was varied from its current weight. Sensitivity analysis can be performed using either a change to Local Proportional or Global Proportional weightings.

Value Scores

The spreadsheet shown in Figure 70 displays the en route alternatives in rank-order according to their total Tactical Submodel score. The row shaded grey lists the global weightings for each measure, thus this value is also the maximum achievable score for that measure.

		Origin Ramstein AB, GE (ETAR)		Destination Hosea Kutako Intl, Namibia (FYWH)		ALL MEASURES																									
RANK	Best Possible Score	TOTAL SCORE	MOG	Critical Leg	Delta Flight Length	Alternate Airfields	Fuel Storage	Fuel Resupply	Dip Clearance	Force Protection	Dept of State	Military Cooperation	Seaport	Railroad	Road System	Commercial Airport	Lodging	Dining	Medical	Communications	Power	Potable Water	Sewer	Mountains	Altitude	Weather	Temperature	Urban Areas	Terrain		
1	Charley and Alex Rule	0.942	0.9421	0.0715	0.0475	0.0714	0.0230	0.0069	0.0265	0.0374	0.1111	0.0741	0.1111	0.0057	0.0087	0.0217	0.0425	0.0221	0.0161	0.0023	0.0052	0.0359	0.0200	0.0205	0.0349	0.0174	0.0805	0.0171	0.0569	0.0114	
2	*Constanta, Romania (LRCK)	0.782	0.7821	0.0333	0.0680	0.0200	0.0199	0.0068	0.0000	0.0299	0.1111	0.0743	0.1111	0.0080	0.0076	0.0203	0.0435	0.0321	0.0161	0.0032	0.0052	0.0369	0.0206	0.0205	0.0342	0.0171	0.0685	0.0171	0.0569	0.0114	
3	*Bucharest, Romania (LROP)	0.725	0.7248	0.0333	0.0680	0.0000	0.0000	0.0007	0.0047	0.0299	0.1111	0.0743	0.1111	0.0023	0.0023	0.0161	0.0032	0.0052	0.0278	0.0185	0.0185	0.0342	0.0171	0.0274	0.0171	0.0569	0.0114				
4	Dakar, Senegal (GOOY)	0.721	0.7206	0.0306	0.0308	0.0240	0.0025	0.0069	0.0023	0.0224	0.1111	0.0741	0.1111	0.0085	0.0087	0.0066	0.0210	0.0435	0.0321	0.0161	0.0032	0.0046	0.0278	0.0185	0.0185	0.0342	0.0171	0.0685	0.0171	0.0000	0.0000
5	Kotoka Intl, Ghana (DGAA)	0.692	0.6916	0.0114	0.0418	0.0217	0.0571	0.0043	0.0002	0.0161	0.0224	0.0778	0.0741	0.0089	0.0087	0.0087	0.0210	0.0435	0.0203	0.0161	0.0032	0.0046	0.0162	0.0000	0.0000	0.0342	0.0171	0.0685	0.0171	0.0569	0.0114
6	Thumrait, Oman, (OOTH)	0.684	0.6838	0.0171	0.0217	0.0048	0.0195	0.0056	0.0008	0.0073	0.0299	0.1111	0.0722	0.0087	0.0089	0.0089	0.0203	0.0435	0.0257	0.0161	0.0032	0.0046	0.0278	0.0185	0.0185	0.0342	0.0171	0.0685	0.0171	0.0569	0.0114
7	Nairobi Jkia, Kenya (HKJK)	0.652	0.6525	0.0223	0.0000	0.0000	0.0000	0.0000	0.0000	0.0224	0.0224	0.0443	0.0443	0.0089	0.0089	0.0089	0.0224	0.0435	0.0257	0.0161	0.0032	0.0052	0.0278	0.0185	0.0185	0.0342	0.0171	0.0685	0.0171	0.0569	0.0114
8	*Burgas, Bulgaria (LBBG)	0.643	0.6287	0.0091	0.0091	0.0059	0.0059	0.0219	0.0299	0.0630	0.0222	0.0556	0.0556	0.0089	0.0089	0.0089	0.0219	0.0435	0.0257	0.0161	0.0032	0.0052	0.0278	0.0186	0.0186	0.0342	0.0171	0.0685	0.0171	0.0569	0.0114
9	Cairo West, Egypt (HECW)	0.629	0.6286	0.0048	0.0202	0.0459	0.0000	0.0008	0.0014	0.0299	0.0778	0.0222	0.0889	0.0333	0.0053	0.0196	0.0289	0.0193	0.0088	0.0032	0.0046	0.0278	0.0186	0.0186	0.0342	0.0171	0.0685	0.0171	0.0569	0.0114	
10	*Incirlik AB, Turkey (LTAG)	0.622	0.6222	0.0061	0.0000	0.0178	0.0007	0.0003	0.0043	0.0143	0.0224	0.0778	0.0222	0.1111	0.0087	0.0087	0.0210	0.0399	0.0321	0.0161	0.0032	0.0052	0.0206	0.0205	0.0342	0.0171	0.0685	0.0171	0.0000	0.0000	
11	*Heydar Aliyev, Azerbaijan (UBBJ)	0.604	0.6036	0.0083	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0556	0.0556	0.0078	0.0076	0.0076	0.0189	0.0336	0.0321	0.0161	0.0032	0.0052	0.0308	0.0185	0.0185	0.0342	0.0171	0.0685	0.0171	0.0569	0.0114
12	Al Udeid AB, Qatar (OTIH)	0.591	0.5906	0.0114	0.0208	0.0025	0.0132	0.0037	0.0000	0.0299	0.0778	0.0222	0.0809	0.0072	0.0000	0.0000	0.0189	0.0336	0.0321	0.0161	0.0032	0.0052	0.0308	0.0185	0.0185	0.0342	0.0171	0.0685	0.0171	0.0569	0.0114
13	Entebbe Intl, Uganda (HUEN)	0.545	0.5451	0.0036	0.0215	0.0481	0.0000	0.0006	0.0000	0.0224	0.0778	0.0222	0.0556	0.0087	0.0021	0.0183	0.0435	0.0321	0.0161	0.0032	0.0052	0.0165	0.0062	0.0113	0.0113	0.0342	0.0000	0.0274	0.0171	0.0569	0.0000
14	*Bagram, Afghanistan (OAK)	0.496	0.4958	0.0065	0.0000	0.0000	0.0116	0.0038	0.0161	0.0299	0.0000	0.0445	0.0089	0.0000	0.0000	0.0108	0.0305	0.0257	0.0165	0.0032	0.0046	0.0278	0.0185	0.0185	0.0342	0.0171	0.0685	0.0171	0.0569	0.0000	
15	Djibouti Ambouli, Djibouti (HDAM)	0.397	0.3967	0.0049	0.0061	0.0261	0.0000	0.0009	0.0224	0.0222	0.0556	0.0062	0.0076	0.0210	0.0435	0.0000	0.0000	0.0015	0.0062	0.0113	0.0113	0.0342	0.0171	0.0685	0.0000	0.0000	0.0000	0.0000	0.0000		

Figure 70: Value Scores Spreadsheet

Color formatting is used to highlight score values achieved by the alternatives. A score formatted **RED** indicates that the alternative achieved no score in that measure. A score formatted **TEAL** indicates that the alternative achieved the highest possible score for that measure.

Additionally, alternatives are tagged according to screening criteria that may cause a user to filter the alternative from the feasible alternatives list.

- * indicates the alternative is not within the lens for the given origin-destination pair
- # indicates the Diplomatic Clearance measure is “None”
- ! indicates the Dept of State measure is “Unacceptable”

Rankings Chart

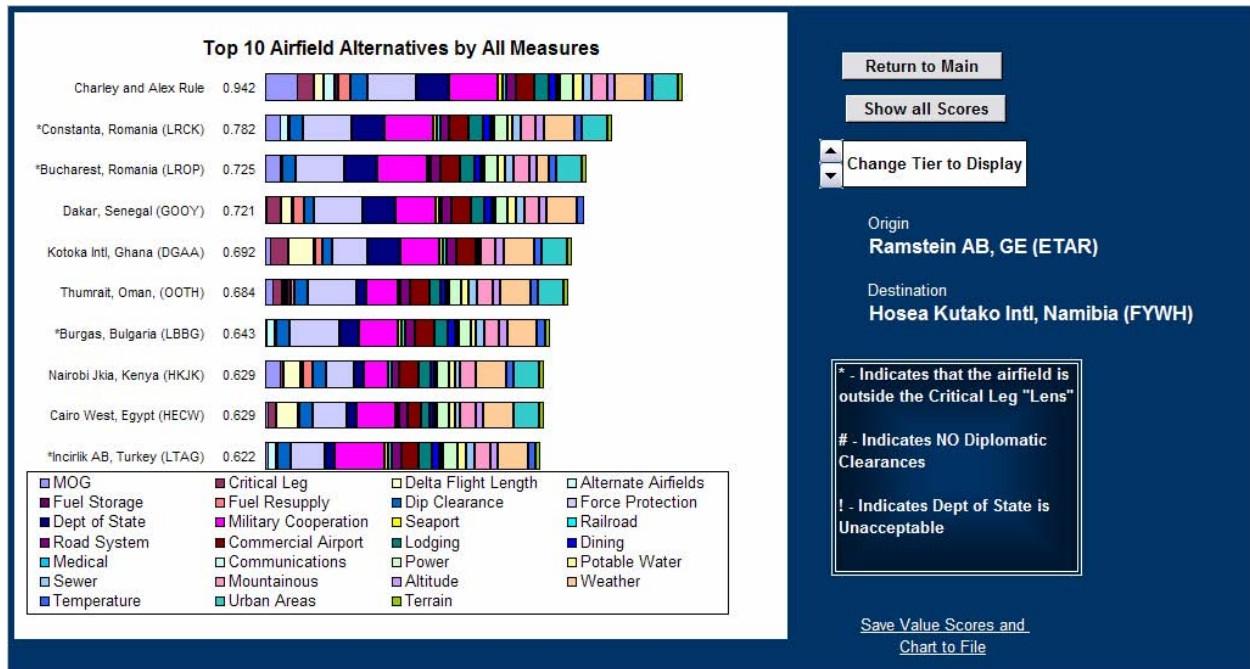


Figure 71: Tactical Score Rankings Chart

The rankings chart displays the top 10 ranking alternatives, regardless of screening criteria. Alternative labels are tagged with screening remarks to let the user decide whether to filter the alternative or not.

Displaying by different Tiers

The rankings chart can be rolled up/down through various tiers. This means the overall score bar chart display can be changed to be broken down by varying tiers of objectives. For example, choosing to chart by the Tier 1 Values will break down alternatives' tactical scores by the Throughput Score and Sustainment Score only. Pressing on the spinner button to **Change Tier to Display** scrolls the display up or down and allows the chart to be displayed by either Tier 1, Tier 2, Tier 3, or all measures.

Copying Value/Rank Chart Results to File

Selecting Save Value Scores and Chart to File from either the Value Scores spreadsheet or the Rankings chart will send a copy of both the Value Scores spreadsheet the Rankings Chart to a separate results file.

The result file is named GERBIL Model Results.xls and can be found in the user's designated MyDocuments folder.

All links from the program application are disconnected and another top 10 chart, displaying results from all 27 measures, along with a spreadsheet, displaying all en route alternatives and measure scores, is saved to the worksheet mentioned above.

Copying results to a separate file allows the user to save results and modify formatting for presentation if desired.

Sensitivity Analysis

Displays a line chart depicting how the results would vary if the weighting for the selected objective was varied from its current weight. Sensitivity analysis can be performed using either Local Proportional or Global Proportional weighting.

Sensitivity analysis is useful in demonstrating the robustness of the overall results by displaying the effect of changing the level of importance of a particular objective.

Each line shows the Tactical Submodel score of an en route alternative as the weighting of the selected objective is varied from its current weight, the red line.

Only en route alternatives that are within the critical leg lens are displayed on the sensitivity analysis chart.

Local Proportional

Local Proportional sensitivity analysis varies the weight of the selected objective from 0 to 100% of the score within its branch, while maintaining the original proportions for the other objectives on its branch. (A branch is a set of objectives on the same tier with the same parent).

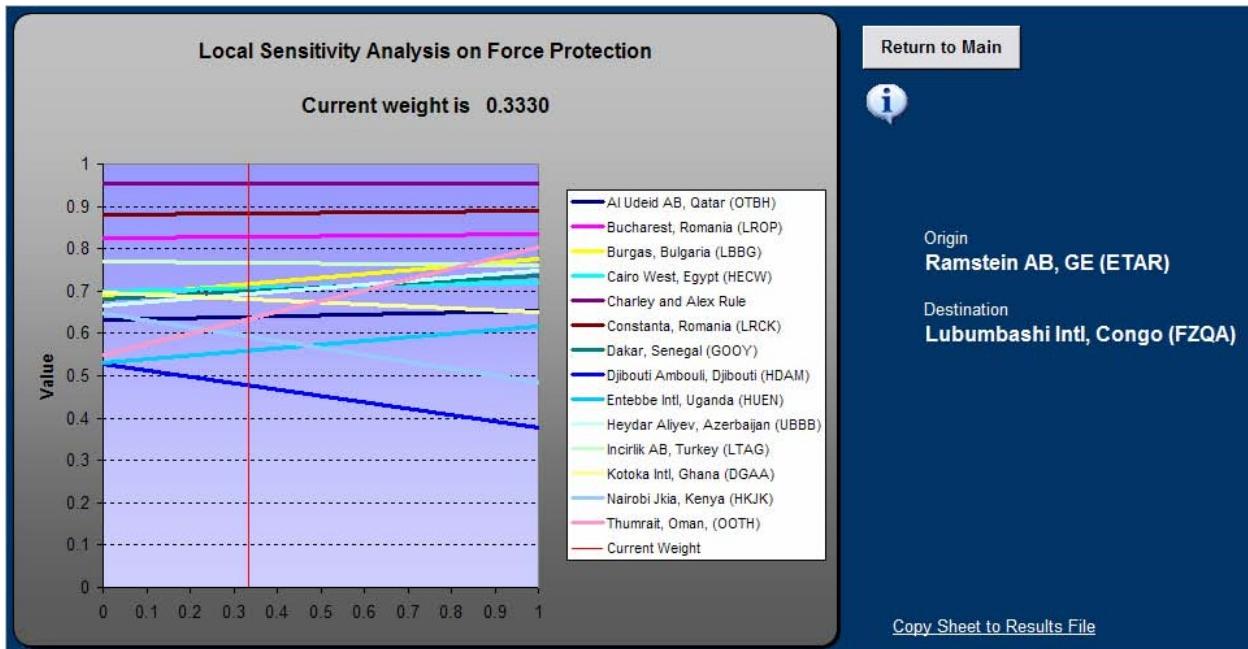


Figure 72: Sample Local Proportional Sensitivity Analysis

Global Proportional

Global Proportional sensitivity analysis varies the weight of the selected objective from 0 to 100% of the overall score while maintaining the original proportion of all other objectives.

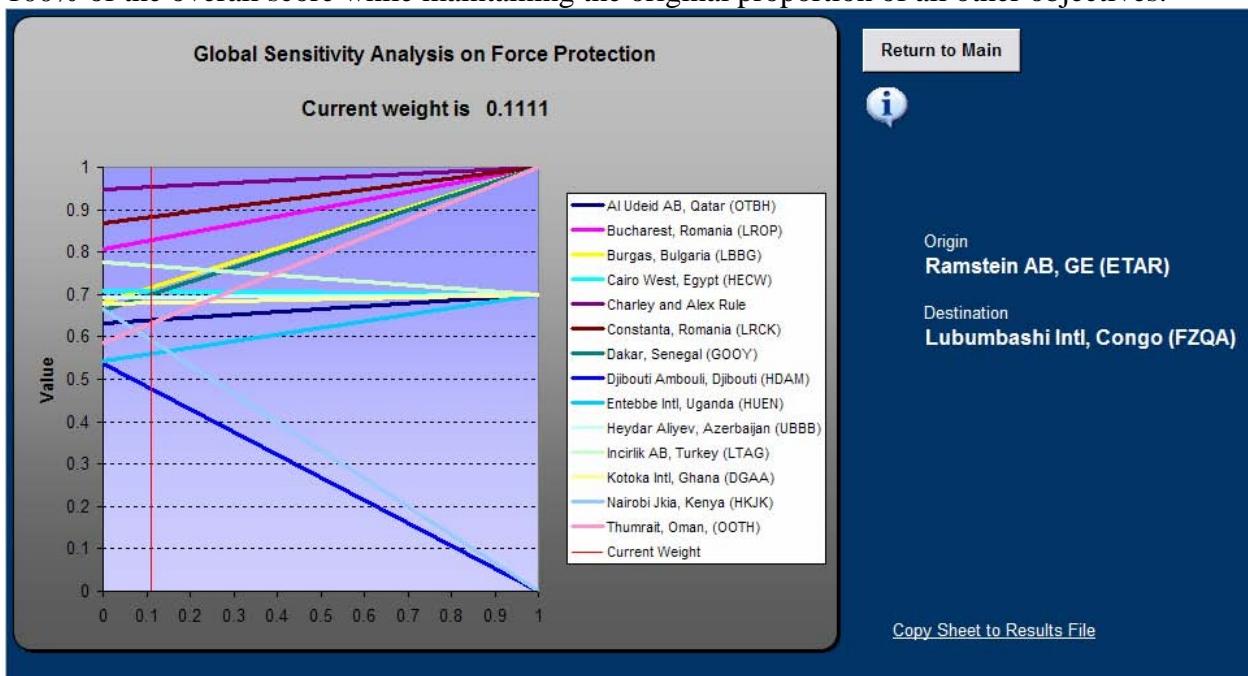


Figure 73: Sample Global Proportional Sensitivity Analysis

Copying Sensitivity Analysis Results to File

Selecting Copy Sheet to Results File will store the results to a separate file.

The result file is named GERBIL Model Results.xls and can be found in the user's designated MyDocuments folder.

The links from the program application are disconnected. The scores used as source data to draw the chart can be found in row 5000 of the results worksheet.

Copying results to a separate file allows the user to save results and modify formatting for presentation if desired.

Running an Operational Scenario

This module is used to apply the Operational Value Hierarchy Model. An operational scenario is used to determine the en route alternatives that are valued the most given a specific origin base and a list of destinations that the user desires to access.

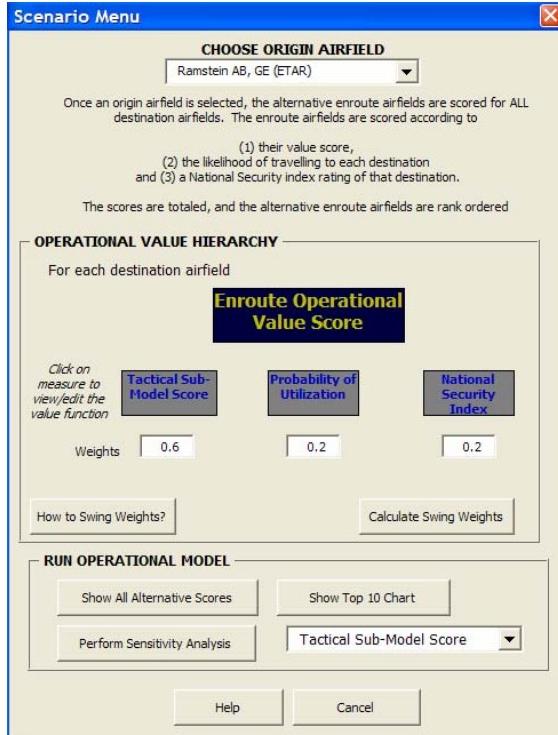


Figure 74: Operational Scenario Menu

In this module, the user will specify the origin base of interest.

For each destination, each alternative will be evaluated according to its Tactical Submodel score, its Probability of Utilization to access the destination region, and the level of National Security Import for the destination region it is accessing.

Each alternative accumulates value for each destination it can access.

The top 10 en route alternatives for the given origin airfield can be displayed in a rankings bar chart. Each en route score can also be viewed in spreadsheet format.

The value functions and weighting for the Operational Value Hierarchy can be viewed and modified. Sensitivity Analysis on the weighting used in the Operational Value Hierarchy can be performed to determine the robustness of scenario results.

Origin Selection

The origin airfield for the operational scenario is selected from the Operational Scenario Menu list box, not from the list box in the main menu.

Operational Value Hierarchy

The Operational Value Hierarchy consists of one branch of three objectives: Tactical Submodel score (from running the Tactical Submodel), Probability Score, and National Security Import.

Weighting the Operational Value Hierarchy

No original or default weights are saved for the Operational Value Hierarchy. Only the previous set of valid weights are maintained. As in the View Tactical Submodel Module, the weighting for the branch can be changed manually or by using the swing weighting technique. For a tutorial on swing weighting, the user can click on How to Swing Weights?.

Swing Weighting

See the discussion on Swing Weighting in the View Tactical Submodel Module section.

Value Functions

There is no value function for the Tactical Submodel score. The score is taken directly from the score of running the Tactical Submodel for each origin-destination pairing.

Probability of Utilization and the Prospect Function

The Probability of Utilization of a destination airfield to access the destination region is achieved by utilizing The Prospect Theory developed by Tversky and Kahneman (1992). The Prospect Theory takes into account the tendency for humans to weight small changes in probability at the extremes more than the equivalent numerical change in the middle. This value function cannot be modified in GERBIL.

$$\pi(P_d) = 1.89799P_d - 3.55955P_d^2 + 2.662549P_d^3$$

P_d	Probability of Utilization of the destination airfield to access the destination region, as elicited from the DM
$\pi(P_d)$	Value given for a specific probability, P_d

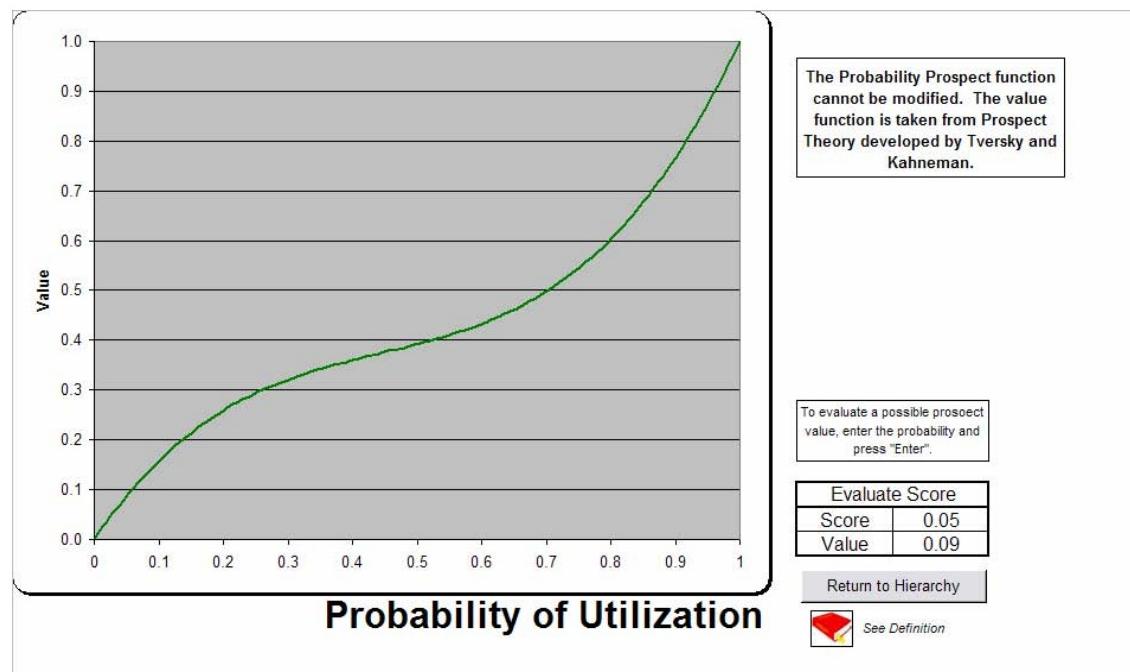


Figure 75: Prospect Function for Probability of Utilization

National Security Index Function

The National Security SDVF is the only function in the Operational Value Hierarchy that can be modified as in the View Tactical Submodel Module. The National Security SDVF is a continuous increasing value function. For more information on editing this function, see Continuous Increasing Value Functions in the View Tactical Submodel Module section.

Operational Scenario Results

Running the Operational Value Hierarchy Model yields three different types of results. A Scenario Score Matrix can be displayed showing the total score of each en route alternative broken down by destinations. A Scenario Rankings chart can be displayed showing the top 10 ranking alternatives. Finally, a Sensitivity Analysis can be performed on the weighting of any of the three objectives.

Scenario Scores

Selecting Show All Alternative Scores displays a spreadsheet of the operational value scores for each alternative. The score for each en route alternative airfield is broken down by each of the destinations.

If the distance between an origin-destination pair is less than the maximum critical leg, then NO ENROUTE IS REQUIRED, and no en route alternatives are scored for accessing that destination from the origin. This situation is indicated by formatting the destination column in italicized **GREY**.

When origin-destination pairs require an en route airfield, and an en route alternative is not within the lens, it will receive a score of zero, displayed in **RED**, for accessing that destination.

		GREY - Indicates enroute airfield not required for origin-destination pair		ORIGIN		Moron AB, SP (LEMO)											
		Destination Airfields															
		Diego Garcia NSF, Chhatrapati Shivaji International, India (VABB), British Indian Ocean Territory (FJDG), Arkonam, India (VOAR), Bandaranaike International, Columbol, Sri Lanka (VCBI), Tribhuvan Intl, Nepal (VNKT)															
Rank	Alternatives	TOTAL	Hosea Kutako Intl, Namibia (FYWH)	Cape Town Intl, South Africa, (FACT)	Lubumbashi Intl, Congo (FZQA)	Lusaka International, Zambia (FLLS)	Dar Es Salaam, Tanzania (HTDA)	Chhatrapati Shivaji International, India (VABB)	Diego Garcia NSF, British Indian Ocean Territory (FJDG)	Arkonam, India (VOAR)	Bandaranaike Int'l, Columbol, Sri Lanka (VCBI)	Tri	bandhuvan	Intl, Nepal (VNKT)			
1	Charley and Alex Rule	4.629	0.667	0.673	0	0	0	0.650	0.747	0.653	0.643	0.598					
2	Nairobi Jkia, Kenya (HKJK)	3.270	0.497	0.519	0	0	0	0.440	0.550	0.445	0.436	0.383					
3	Thumrait, Oman, (OOTH)	3.139	0.516	0	0	0	0	0.519	0.608	0.521	0.510	0.465					
4	Cairo West, Egypt (HECW)	2.907	0.491	0	0	0	0	0.485	0.554	0.480	0.463	0.434					
5	Al Udeid AB, Qatar (OTBH)	2.877	0.458	0	0	0	0	0.480	0.561	0.481	0.467	0.431					
6	Entebbe Intl, Uganda (HUEN)	2.836	0.443	0.464	0	0	0	0.379	0.487	0.380	0.372	0.311					
7	Heydar Aliyev, Azerbaijan (UBBB)	2.534	0	0	0	0	0	0.506	0.567	0.505	0.487	0.469					
8	Incirlik AB, Turkey (LTAG)	2.534	0	0	0	0	0	0.509	0.567	0.505	0.486	0.466					
9	Djibouti Ambouli, Djibouti (HDAM)	2.345	0.344	0.374	0	0	0	0.316	0.417	0.320	0.310	0.264					
10	Constanta, Romania (LRCK)	2.232	0	0	0	0	0	0.581	0	0.570	0.542	0.540					
11	Bagram, Afghanistan (OAIX)	2.002	0	0	0	0	0	0.396	0.465	0.397	0.380	0.364					
12	Burgas, Bulgaria (LBBG)	1.967	0	0	0	0	0	0.515	0	0.504	0.476	0.472					
13	Bucharest, Romania (LROP)	1.580	0	0	0	0	0	0.545	0	0.531	0	0.504					
14	Kotoka Intl, Ghana (DGAA)	1.073	0.528	0.545	0	0	0	0	0	0	0	0					
15	Dakar, Senegal (GOOY)	0.539	0.539	0	0	0	0	0	0	0	0	0					

Figure 76: Scenario Score Matrix

Scenario Top 10 Chart

Selecting Show Top 10 Chart displays a bar chart for the en route alternatives that score in the top 10 of the Operational Scenario. A bar is indicated for any destination airfield that an en route alternative can access.

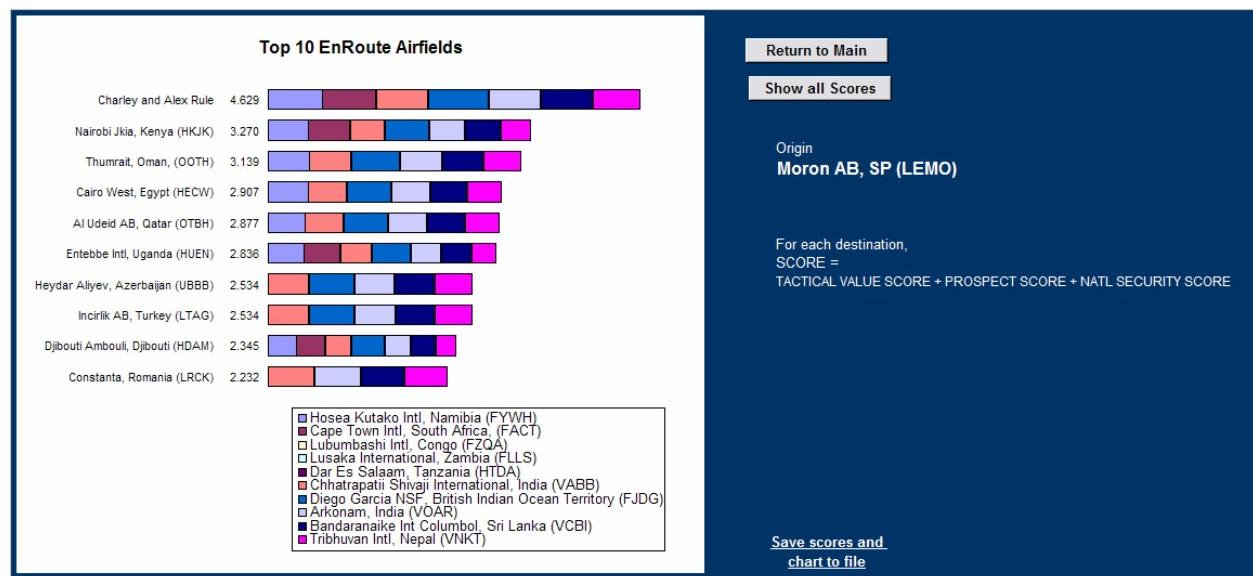


Figure 77: Scenario Rankings Chart

Copying Scores/Chart Results to File

Selecting Save scores and chart to file from either the Scenario Scores Matrix spreadsheet or the Scenario Rankings chart sends both the Scenario Scores Matrix spreadsheet and the rankings chart to a separate results file.

The result file is named GERBIL Model Results.xls and can be found in the user's designated MyDocuments folder.

The links from the program application are disconnected and another top 10 chart display is drawn in the saved worksheet.

Copying results to a separate file allows the user to save results and modify formatting for presentation if desired.

Scenario Sensitivity Analysis

Selecting Perform Sensitivity Analysis displays a line chart of the scores for all alternatives as the weight of the objective selected from the adjacent list box is varied from 0 to 1. The original proportions for the remaining objectives are maintained.

Sensitivity analysis is useful in demonstrating the robustness of the overall results by displaying the effect of changing the level of importance of a particular objective.

Each line shows the Operational Value Score of an en route alternative as the weighting of the selected objective is varied from its current weight, the red line.

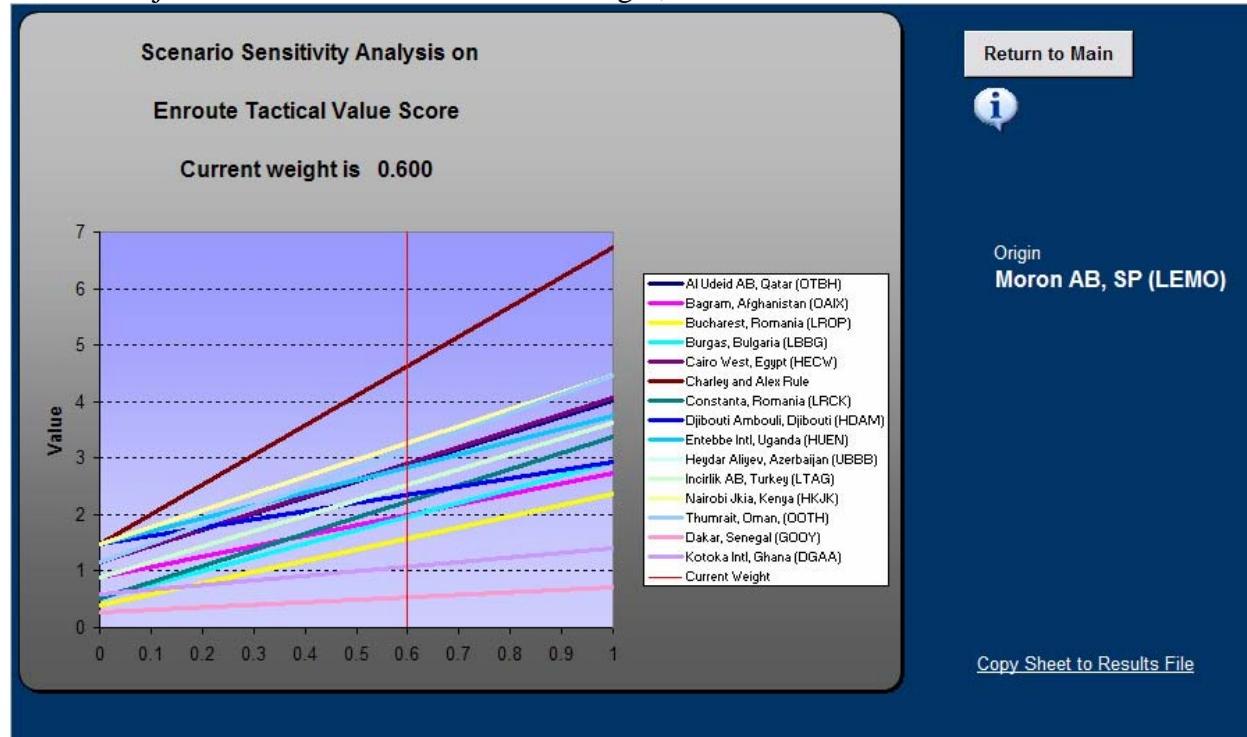


Figure 78: Sample Scenario Sensitivity Analysis Chart

Copying Sensitivity Analysis Results to File

Selecting Copy Sheet to Results File saves the results to a separate file.

The result file is named GERBIL Model Results.xls and can be found in the user's designated MyDocuments folder.

The links from the program application are disconnected. The scores used as source data to draw the chart can be found in row 5000 of the results worksheet.

Copying results to a separate file allows the user to save results and modify formatting for presentation if desired.

Appendix 3. GERBIL Visual Basic Code

The following pages contain the visual basic code for GERBIL Version 1.0

Charts/Worksheets/Workbook Code

User Forms Code

Modules Code

Charts/Worksheets/Workbook Code

```
Chart1 - 1
Private Sub Chart_Activate()
ActiveChart.Unprotect
ActiveChart.Axes(xlCategory).Select
Selection.TickLabels.Font.Size = 7.75
ActiveChart.Protect
End Sub

Sheet1 - 1
Private Sub Worksheet_Activate()
ActiveSheet.Shapes("Gerbil Quote").Visible = False
End Sub

Sheet10 - 1
Private Sub Worksheet_Activate()
With ActiveWindow
.DisplayHorizontalScrollBar = True
.DisplayVerticalScrollBar = True
End With
End Sub
Private Sub Worksheet_Deactivate()
With ActiveWindow
.DisplayHorizontalScrollBar = False
.DisplayVerticalScrollBar = False
End With
End Sub

Sheet4 - 1
Private Sub Worksheet_Activate()
With ActiveWindow
.DisplayHorizontalScrollBar = True
.DisplayVerticalScrollBar = True
.ScrollColumn = 1
.ScrollRow = 10
End With
ScrollArea = "A:Y"
End Sub
Private Sub Worksheet_Deactivate()
With ActiveWindow
.DisplayHorizontalScrollBar = False
.DisplayVerticalScrollBar = False
End With
End Sub

Sheet5 - 1
Private Sub Worksheet_Activate()
With ActiveWindow
.DisplayHorizontalScrollBar = True
.DisplayVerticalScrollBar = True
.ScrollColumn = 1
.ScrollRow = 1
End With
ScrollArea = "A:AG"
End Sub
Private Sub Worksheet_Deactivate()
With ActiveWindow
.DisplayHorizontalScrollBar = False
.DisplayVerticalScrollBar = False
End With
End Sub
```

```

Sheet7 - 1
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
End Sub

ThisWorkbook - 1
Private Sub Workbook_Activate()
With ActiveWindow
    .DisplayHeadings = False
    .DisplayOutline = False
    .DisplayHorizontalScrollBar = False
    .DisplayVerticalScrollBar = False
End With
With Application
    .ShowStartupDialog = False
    .DisplayFormulaBar = False
    .ShowWindowsInTaskbar = False
End With
End Sub

Private Sub Workbook_BeforeClose(Cancel As Boolean)
Application.ScreenUpdating = False
' Delete links to avoid error message
Worksheets("Sensitivity").ChartObjects.Delete
Worksheets("Rankings").ChartObjects.Delete
Worksheets("Scenario Sensitivity").ChartObjects.Delete
Worksheets("Scenario Rankings").ChartObjects.Delete
Worksheets("Main").Visible = True
Worksheets("Main").Activate
Worksheets("Rankings").Visible = False
Worksheets("Scenario Rankings").Visible = False
Application.ScreenUpdating = False
End Sub

Private Sub Workbook_Deactivate()
With ActiveWindow
    .DisplayHeadings = True
    .DisplayOutline = True
    .DisplayHorizontalScrollBar = True
    .DisplayVerticalScrollBar = True
End With
With Application
    .ShowStartupDialog = True
    .DisplayFormulaBar = True
    .ShowWindowsInTaskbar = True
End With
End Sub

Private Sub Workbook_Open()
Application.ScreenUpdating = False
' Call UnprotectProgram
' Set default Origin and Destination fields
' Worksheets("Orig-Dest Airfields").Range("B1") = 8
' Worksheets("Orig-Dest Airfields").Range("B2") = 1
Call ApplicationInterface
Application.ScreenUpdating = True
End Sub

```

User Forms Code

```
AddEditDestination = 1
Private Sub Airfield_Change()
End Sub
Private Sub CheckBox1_Click()
If CheckBox1 = True Then
    NationalSecurityMode.Value = NationalSecurityMin.Value
    NationalSecurityMin.Value = 1
    NationalSecurityMax.Value = 10
    Label46.Visible = True
    Label47.Visible = True
    Label48.Visible = True
    NationalSecurityMode.Visible = True
    NationalSecurityMax.Visible = True
    NationalSecurityMode.Enabled = True
    NationalSecurityMax.Enabled = True
Else
    NationalSecurityMin.Value = NationalSecurityMode.Value
    Label46.Visible = False
    Label47.Visible = False
    Label48.Visible = False
    NationalSecurityMode.Visible = False
    NationalSecurityMax.Visible = False
    NationalSecurityMode.Enabled = False
    NationalSecurityMax.Enabled = False
End If
End Sub
Private Sub CommandButton1_Click()
ActiveSheet.Unprotect
If AddEditDestination.Caption = "Add Destination Airfield" Then
    NextRow = Application.WorksheetFunction.CountA(Range("K11:K65356")) + FirstOD
Else
    NextRow = FirstOD + AddEditDestination.Index
End If
With Worksheets("Orig-Dest Airfields")
    .Cells(NextRow, 11) = Airfield.Value
    .Cells(NextRow, 12) = Probability.Value / 100
    .Cells(NextRow, 12).NumberFormat = "0.00"
    If CheckBox1 = True Then
        .Cells(NextRow, 13) = "?"
        .Cells(NextRow, 14) = NationalSecurityMin.Value
        .Cells(NextRow, 15) = NationalSecurityMode.Value
        .Cells(NextRow, 16) = NationalSecurityMax.Value
    Else
        .Cells(NextRow, 13) = ""
        .Cells(NextRow, 14) = NationalSecurityMin.Value
        .Cells(NextRow, 15) = NationalSecurityMin.Value
        .Cells(NextRow, 16) = NationalSecurityMin.Value
    End If
    .Cells(NextRow, 14).NumberFormat = "0"
    .Cells(NextRow, 15).NumberFormat = "0"
    .Cells(NextRow, 16).NumberFormat = "0"
    ' .Cells(NextRow, 17) = Throughput.Value
    .Cells(NextRow, 18) = NorS.Value
    .Cells(NextRow, 19) = LatDeg.Value
    .Cells(NextRow, 20) = LatMin.Value
    .Cells(NextRow, 20).NumberFormat = "0.00"
    .Cells(NextRow, 21) = EorW.Value
    .Cells(NextRow, 22) = LongDeg.Value
    .Cells(NextRow, 23) = LongMin.Value
    .Cells(NextRow, 23).NumberFormat = "0.00"
End With

```

```

.Cells(NextRow, 24).NumberFormat = "0.00"
.Cells(NextRow, 25).NumberFormat = "0.00"
'Evaluate the Inverse Prospect Value for the given Probability
.Cells(NextRow, 30) = ProspectVal(Probability.Value / 100)
AddEditDestination - 2
End With
NextRowRange = "X" & Application.WorksheetFunction.Text(NextRow, 0)
Range("X11:Y11").Copy Range(NextRowRange)
Call SortOrigDest
Unload Me
End Sub
Private Sub CommandButton2_Click()
Unload Me
End Sub
Private Sub Label48_Click()
End Sub
Private Sub LatDeg_AfterUpdate()
If LatDeg.Value < 0 Or LatDeg.Value > 75 Then
MsgBox "Value must be between 0 and 75", vbOKOnly, "Invalid Input"
LatDeg.Value = 0
LatDeg.SetFocus
End If
End Sub
Private Sub LatMin_AfterUpdate()
If LatMin.Value < 0 Or LatMin.Value >= 60 Then
MsgBox "Value must be between 0.00 and 59.99", vbOKOnly, "Invalid Input"
LatMin.Value = 0
LatMin.SetFocus
End If
End Sub
Private Sub Probability_AfterUpdate()
If Probability.Value < 0 Or Probability.Value > 100 Then
MsgBox "Value must be between 0 and 100", vbOKOnly, "Invalid Input"
Probability.Value = 0
Probability.SetFocus
End If
End Sub
Private Sub LongDeg_AfterUpdate()
If LongDeg.Value < 0 Or LongDeg.Value > 180 Then
MsgBox "Value must be between 0 and 180", vbOKOnly, "Invalid Input"
LongDeg.Value = 0
LongDeg.SetFocus
End If
End Sub
Private Sub LongMin_AfterUpdate()
If LongMin.Value < 0 Or LongMin.Value >= 60 Then
MsgBox "Value must be between 0.00 and 59.99", vbOKOnly, "Invalid Input"
LongMin.Value = 0
LongMin.SetFocus
End If
End Sub
Private Sub NationalSecurityMin_AfterUpdate()
If NationalSecurityMin.Value < 1 Or NationalSecurityMin.Value > 10 Then
MsgBox "Value must be between 1 and 10", vbOKOnly, "Invalid Input"
NationalSecurityMin.Value = 1
NationalSecurityMin.SetFocus
End If
End Sub
Private Sub NationalSecurityMode_AfterUpdate()
If NationalSecurityMode.Value < 1 Or NationalSecurityMode.Value > 10 Then
MsgBox "Value must be between 1 and 10", vbOKOnly, "Invalid Input"
NationalSecurityMode.Value = 5
AddEditDestination - 3
NationalSecurityMode.SetFocus

```

```

End If
End Sub
Private Sub NationalSecurityMax_AfterUpdate()
If NationalSecurityMax.Value < 1 Or NationalSecurityMax.Value > 10 Then
MsgBox "Value must be between 1 and 10", vbOKOnly, "Invalid Input"
NationalSecurityMax.Value = 5
NationalSecurityMax.SetFocus
End If
End Sub
Private Sub Throughput_AfterUpdate()
If Throughput.Value < 0 Then
MsgBox "Value must be greater than 0", vbOKOnly, "Invalid Input"
Throughput.Value = 0
Throughput.SetFocus
End If
End Sub
Private Sub UserForm_Initialize()
' Initialize Lat/Long for N or S
NorS.AddItem "N"
NorS.AddItem "S"
EorW.AddItem "E"
EorW.AddItem "W"
NorS.ListIndex = 0 ' North
EorW.ListIndex = 0 ' East
LatDeg.Value = 0
LatMin.Value = 0
LongDeg.Value = 0
LongMin.Value = 0
Probability.Value = 0
NationalSecurityMin.Value = 5
NationalSecurityMode.Value = 5
NationalSecurityMax.Value = 5
'Throughput.Value = 0
CheckBox1 = False
Label46.Visible = False
Label47.Visible = False
Label48.Visible = False
NationalSecurityMode.Visible = False
NationalSecurityMax.Visible = False
NationalSecurityMode.Enabled = False
NationalSecurityMax.Enabled = False
End Sub
Function ProspectVal(Probability)
'Uses Goal Seek to evaluate the inverse prospect function
Range("Q3").GoalSeek Goal:=Probability, ChangingCell:=Range("Q4")
ProspectVal = Cells(4, 17)
End Function

AddOrigin - 1
Private Sub Airfield_Change()
End Sub
Private Sub CommandButton1_Click()
ActiveSheet.Unprotect
NextRow = Application.WorksheetFunction.CountA(Range("A11:A65356")) + FirstOD
With Worksheets("Orig-Dest Airfields")
.Cells(NextRow, 1) = Airfield.Value
.Cells(NextRow, 2) = NorS.Value
.Cells(NextRow, 3) = LatDeg.Value
.Cells(NextRow, 4) = LatMin.Value
.Cells(NextRow, 4).NumberFormat = "0.00"
.Cells(NextRow, 5) = EorW.Value
.Cells(NextRow, 6) = LongDeg.Value
.Cells(NextRow, 7) = LongMin.Value
.Cells(NextRow, 7).NumberFormat = "0.00"

```

```

.Cells(NextRow, 9).NumberFormat = "0.00"
.Cells(NextRow, 10).NumberFormat = "0.00"
End With
NextRowRange = "H" & Application.WorksheetFunction.Text(NextRow, 0)
Range("H11:I11").Copy Range(NextRowRange)
Call SortOrigDest
Unload Me
End Sub
Private Sub CommandButton2_Click()
Unload Me
End Sub
Private Sub LatDeg_AfterUpdate()
If LatDeg.Value < 0 Or LatDeg.Value > 75 Then
MsgBox "Value must be between 0 and 75", vbOKOnly, "Invalid Input"
LatDeg.Value = 0
LatDeg.SetFocus
End If
End Sub
Private Sub LatMin_AfterUpdate()
If LatMin.Value < 0 Or LatMin.Value >= 60 Then
MsgBox "Value must be between 0.00 and 59.99", vbOKOnly, "Invalid Input"
LatMin.Value = 0
LatMin.SetFocus
End If
End Sub
Private Sub LongDeg_AfterUpdate()
If LongDeg.Value < 0 Or LongDeg.Value > 180 Then
MsgBox "Value must be between 0 and 180", vbOKOnly, "Invalid Input"
LongDeg.Value = 0
LongDeg.SetFocus
End If
End Sub
Private Sub LongMin_AfterUpdate()
If LongMin.Value < 0 Or LongMin.Value >= 60 Then
MsgBox "Value must be between 0.00 and 59.99", vbOKOnly, "Invalid Input"
LongMin.Value = 0
LongMin.SetFocus
End If
End Sub
Private Sub UserForm_Initialize()
' Initialize Lat/Long for N or S
NorS.AddItem "N"
NorS.AddItem "S"
AddOrigin = 2
EorW.AddItem "E"
EorW.AddItem "W"
NorS.ListIndex = 0 ' North
EorW.ListIndex = 0 ' East
End Sub

Aircraft = 1
Dim Weights(4), OldWeights(4)
Dim i As Byte
Private Sub CommandButton1_Click()
HowToSwing.Show
End Sub
Private Sub CommandButton2_Click()
Total = 0
' Swing the Weights
For i = 1 To 4
Total = Total + Weights(i)
Next i
For i = 1 To 4
Weights(i) = Weights(i) / Total

```

```

Next i
TextBox1.Value = Round(Weights(1), 4)
TextBox2.Value = Round(Weights(2), 4)
TextBox3.Value = Round(Weights(3), 4)
TextBox4.Value = Round(Weights(4), 4)
With Sheet3
    .Range("B21") = Weights(1)
    .Range("C21") = Weights(2)
    .Range("D21") = Weights(3)
    .Range("E21") = Weights(4)
End With
Label8.Caption = Round(Sheet3.Range("B22"), 4) 'Mountainous cell B22
Label9.Caption = Round(Sheet3.Range("C22"), 4) 'Altitude cell C22
Label10.Caption = Round(Sheet3.Range("D22"), 4) 'Weather cell D22
Label11.Caption = Round(Sheet3.Range("E22"), 4) 'Climate cell E22
End Sub
Private Sub CommandButton3_Click() ' Accept the new weights if valid
Dim Valid As Boolean
Valid = True
Total = 0
For i = 1 To 4
    Valid = Valid And (Weights(i) <> -1)
    Total = Total + Weights(i)
Next i
If (Not (Valid)) Or Not (EqualOne(Total)) Then
    MsgBox "Please check to ensure all weights are numerical and sum to one." &
        vbCrLf & "This may require 'swinging' the weights.", vbOKOnly, "Invalid Input"
Else:
    With Sheet3
        .Range("B21") = Weights(1)
        .Range("C21") = Weights(2)
        .Range("D21") = Weights(3)
        .Range("E21") = Weights(4)
    End With
Unload Me
End If
End Sub
Private Sub CommandButton4_Click()
Sheet3.Range("B21") = OldWeights(1)
Sheet3.Range("C21") = OldWeights(2)
Sheet3.Range("D21") = OldWeights(3)
Sheet3.Range("E21") = OldWeights(4)
Unload Me
End Sub
Private Sub TextBox1_Change()
Weights(1) = ValidateWt(TextBox1.Value)
End Sub
Private Sub TextBox2_Change()
Weights(2) = ValidateWt(TextBox2.Value)
Aircraft = 2
End Sub
Private Sub TextBox3_Change()
Weights(3) = ValidateWt(TextBox3.Value)
End Sub
Private Sub TextBox4_Change()
Weights(4) = ValidateWt(TextBox4.Value)
End Sub
Private Sub UserForm_Initialize()
    ' Initialize Local Weights
    OldWeights(1) = Sheet3.Range("B21")
    OldWeights(2) = Sheet3.Range("C21")
    OldWeights(3) = Sheet3.Range("D21")
    OldWeights(4) = Sheet3.Range("E21")
    TextBox1.Value = Round(OldWeights(1), 4) 'Mountainous cell B21

```

```

TextBox2.Value = Round(OldWeights(2), 4) 'Altitude cell C21
TextBox3.Value = Round(OldWeights(3), 4) 'Weather cell D21
TextBox4.Value = Round(OldWeights(4), 4) 'Temperature cell E21
' Initialize Global Weights
Label8.Caption = Round(Sheet3.Range("B22"), 4) 'Mountainous cell B22
Label9.Caption = Round(Sheet3.Range("C22"), 4) 'Altitude cell C22
Label10.Caption = Round(Sheet3.Range("D22"), 4) 'Weather cell D22
Label11.Caption = Round(Sheet3.Range("E22"), 4) 'Climate cell E22
For i = 1 To 4
Weights(i) = OldWeights(i)
Next i
End Sub

AirfieldInput - 1
' Variable Definitions for Updating Categorical Airfield Data and the
associated Values
Const DMCOL = 7 'First Col where measure start in Data Sheet
' These variables will be used to post the data to the Airfield Data sheet for
viewing and editing
Dim DipStr, FPStr, DepStateStr, MilCoopStr, LodgingStr, DiningStr, MedStr As
String
Dim CommStr, PowerStr, WaterStr, SewerStr, MountStr, AltStr, WxStr, TempStr As
String
Dim UrbanStr, TerrainStr As String
Private Sub Accept_Click()
' Still need to add a check to validate name and lat/long
'Check to make sure all continuous data inputs are numerical
'Use ValidContData from the Alternatives Module
If Not (ValidContData(MOG.Value) And
ValidContData(AlternateAirfields.Value) And _
ValidContData(FuelStorage.Value) And _
ValidContData(FuelResupply.Value) And _
ValidContData(Seaport.Value) And _
ValidContData(Railroad.Value) And _
ValidContData(RoadSystems.Value) And
ValidContData(CommercialAirport.Value)) Then
MsgBox "One of your continuous data measures is not numerical." & vbCrLf & _
"Please update before clicking Accept.", vbOKOnly, "Invalid Input"
Else
Set USheet = Worksheets("Uncertainty Data")
Set AD = Worksheets("Airfields Data")
USheet.Unprotect
Application.ScreenUpdating = False
USheet.Cells(TempRow, 1) = Airfield.Value
If AirfieldInput.Caption = "Edit airfield alternative" Then
NextRow = AirfieldInput.Index 'The index is preloaded in the
ShowEditAirfieldForm in the Alternatives module
Else
NextRow = Application.WorksheetFunction.CountA(Range("A1:A65356")) + 1 'Else
add a new row
End If
AD.Cells(NextRow, 1) = Airfield.Value
AD.Cells(NextRow, 2) = NorS.Value
AD.Cells(NextRow, 3) = LatDeg.Value + LatMin.Value / 60
AD.Cells(NextRow, 3).NumberFormat = "0.000"
AD.Cells(NextRow, 4) = EorW.Value
AD.Cells(NextRow, 5) = LongDeg.Value + LongMin.Value / 60
AD.Cells(NextRow, 5).NumberFormat = "0.000"
'WriteDataCont and WriteDataCat are in this module
Call WriteDataCont(NextRow, CheckBox1, MOG, 0)
' Critical Leg and Delta Flight Length computed separately
Call WriteDataCont(NextRow, CheckBox2, AlternateAirfields, 3)
Call WriteDataCont(NextRow, CheckBox3, FuelStorage, 4)

```

```

Call WriteDataCont(NextRow, CheckBox4, FuelResupply, 5)
Call WriteDataCat(NextRow, CheckBox5, DipStr, 6)
Call WriteDataCat(NextRow, CheckBox6, FPStr, 7)
Call WriteDataCat(NextRow, CheckBox7, DepStateStr, 8)
Call WriteDataCat(NextRow, CheckBox8, MilCoopStr, 9)
Call WriteDataCont(NextRow, CheckBox9, Seaport, 10)
Call WriteDataCont(NextRow, CheckBox10, Railroad, 11)
Call WriteDataCont(NextRow, CheckBox11, RoadSystems, 12)
Call WriteDataCont(NextRow, CheckBox12, CommercialAirport, 13)
Call WriteDataCat(NextRow, CheckBox13, LodgingStr, 14)
Call WriteDataCat(NextRow, CheckBox14, DiningStr, 15)
Call WriteDataCat(NextRow, CheckBox15, MedStr, 16)
Call WriteDataCat(NextRow, CheckBox16, CommStr, 17)
Call WriteDataCat(NextRow, CheckBox17, PowerStr, 18)
Call WriteDataCat(NextRow, CheckBox18, WaterStr, 19)
Call WriteDataCat(NextRow, CheckBox19, SewerStr, 20)
Call WriteDataCat(NextRow, CheckBox20, MountStr, 21)
Call WriteDataCat(NextRow, CheckBox21, AltStr, 22)
AirfieldInput - 2
Call WriteDataCat(NextRow, CheckBox22, WxStr, 23)
Call WriteDataCat(NextRow, CheckBox23, TempStr, 24)
Call WriteDataCat(NextRow, CheckBox24, UrbanStr, 25)
Call WriteDataCat(NextRow, CheckBox25, TerrainStr, 26)
'Copy Uncertainty Data from TempRow to NextRow, saving it with this airfield
USheet.Range("A3:GN3").Copy USheet.Cells(NextRow + 2, 1)
Call ComputeDistances 'Compute Critical Leg, Delta Flight Length, and Within
the Lens
Call SortFields 'Synchronizes the Airfields Data info with the Uncertainty
Data info
'and sorts according to set criteria (In the Lens then Alphabetical)
Range("A1").Select
Unload Me
Application.ScreenUpdating = True
End If
End Sub
Private Sub AlternateAirfields_Change()
Call CheckCont(AlternateAirfields)
End Sub
Private Sub Altitude_Change()
Call CheckCat(Altitude, AltStr)
End Sub
'=====
=====
' BUTTON CLICK FORMS TO VIEW OR EDIT ASSOCIATED UNCERTAINTY DISTRIBUTIONS
Private Sub Button1_Click()
Call ShowDistForm("MOG")
End Sub
Private Sub Button10_Click()
Call ShowDistForm("Railroad")
End Sub
Private Sub Button11_Click()
Call ShowDistForm("Road System")
End Sub
Private Sub Button12_Click()
Call ShowDistForm("Commercial Airport")
End Sub
Private Sub Button13_Click()
Call ShowDistForm("Lodging")
End Sub
Private Sub Button14_Click()
Call ShowDistForm("Dining")
End Sub
Private Sub Button15_Click()
Call ShowDistForm("Medical")

```

```

End Sub
Private Sub Button16_Click()
Call ShowDistForm("Communications")
End Sub
Private Sub Button17_Click()
Call ShowDistForm("Power")
End Sub
Private Sub Button18_Click()
Call ShowDistForm("Potable Water")
End Sub
Private Sub Button19_Click()
Call ShowDistForm("Sewer")
End Sub
Private Sub Button2_Click()
Call ShowDistForm("Alternate Airfields")
AirfieldInput = 3
End Sub
Private Sub Button20_Click()
Call ShowDistForm("Mountainous")
End Sub
Private Sub Button21_Click()
Call ShowDistForm("Altitude")
End Sub
Private Sub Button22_Click()
Call ShowDistForm("Weather")
End Sub
Private Sub Button23_Click()
Call ShowDistForm("Temperature")
End Sub
Private Sub Button24_Click()
Call ShowDistForm("Urban Areas")
End Sub
Private Sub Button25_Click()
Call ShowDistForm("Terrain")
End Sub
Private Sub Button3_Click()
Call ShowDistForm("Fuel Storage")
End Sub
Private Sub Button4_Click()
Call ShowDistForm("Fuel Resupply")
End Sub
Private Sub Button5_Click()
Call ShowDistForm("Diplomatic Clearance")
End Sub
Private Sub Button6_Click()
Call ShowDistForm("Force Protection")
End Sub
Private Sub Button7_Click()
Call ShowDistForm("Dept of State")
End Sub
Private Sub Button8_Click()
Call ShowDistForm("Military Cooperation")
End Sub
Private Sub Button9_Click()
Call ShowDistForm("Seaport")
End Sub
'=====
=====
=====
Private Sub Cancel_Click()
Unload Me
End Sub
'=====
=====
=====
```

```

=====
'CHECKBOX EVENTS TO ASSIGN PROBABILITY DISTRIBUTIONS
' If checked, the measure input box will be disabled and turned grey
Private Sub CheckBox1_Click()
Const Measure As String = "MOG"
Set MeasureBox = MOG
Call Assign(CheckBox1, MeasureBox, Measure)
End Sub

Private Sub CheckBox10_Click()
Const Measure As String = "Railroad"
Set MeasureBox = Railroad
Call Assign(CheckBox10, MeasureBox, Measure)
End Sub

AirfieldInput = 4
Private Sub CheckBox11_Click()
Const Measure As String = "Road System"
Set MeasureBox = RoadSystems
Call Assign(CheckBox11, MeasureBox, Measure)
End Sub

Private Sub CheckBox12_Click()
Const Measure As String = "Commercial Airport"
Set MeasureBox = CommercialAirport
Call Assign(CheckBox12, MeasureBox, Measure)
End Sub

Private Sub CheckBox13_Click()
Const Measure As String = "Lodging"
Set MeasureBox = Lodging
Call Assign(CheckBox13, MeasureBox, Measure)
End Sub

Private Sub CheckBox14_Click()
Const Measure As String = "Dining"
Set MeasureBox = Dining
Call Assign(CheckBox14, MeasureBox, Measure)
End Sub

Private Sub CheckBox15_Click()
Const Measure As String = "Medical"
Set MeasureBox = Medical
Call Assign(CheckBox15, MeasureBox, Measure)
End Sub

Private Sub CheckBox16_Click()
Const Measure As String = "Communications"
Set MeasureBox = Comm
Call Assign(CheckBox16, MeasureBox, Measure)
End Sub

Private Sub CheckBox17_Click()
Const Measure As String = "Power"
Set MeasureBox = Power
Call Assign(CheckBox17, MeasureBox, Measure)
End Sub

Private Sub CheckBox18_Click()
Const Measure As String = "Potable Water"
Set MeasureBox = Water
Call Assign(CheckBox18, MeasureBox, Measure)
End Sub

Private Sub CheckBox19_Click()
Const Measure As String = "Sewer"
Set MeasureBox = Sewer
Call Assign(CheckBox19, MeasureBox, Measure)
End Sub

Private Sub CheckBox2_Click()
Const Measure As String = "Alternate Airfields"
Set MeasureBox = AlternateAirfields
Call Assign(CheckBox2, MeasureBox, Measure)
End Sub

```

```

Private Sub CheckBox20_Click()
Const Measure As String = "Mountainous"
Set MeasureBox = Mountainous
Call Assign(CheckBox20, MeasureBox, Measure)
End Sub
Private Sub CheckBox21_Click()
Const Measure As String = "Altitude"
Set MeasureBox = Altitude
Call Assign(CheckBox21, MeasureBox, Measure)
End Sub
AirfieldInput - 5
Private Sub CheckBox22_Click()
Const Measure As String = "Weather"
Set MeasureBox = Weather
Call Assign(CheckBox22, MeasureBox, Measure)
End Sub
Private Sub CheckBox23_Click()
Const Measure As String = "Temperature"
Set MeasureBox = Temperature
Call Assign(CheckBox23, MeasureBox, Measure)
End Sub
Private Sub CheckBox24_Click()
Const Measure As String = "Urban Areas"
Set MeasureBox = UrbanAreas
Call Assign(CheckBox24, MeasureBox, Measure)
End Sub
Private Sub CheckBox25_Click()
Const Measure As String = "Terrain"
Set MeasureBox = Terrain
Call Assign(CheckBox25, MeasureBox, Measure)
End Sub
Private Sub CheckBox3_Click()
Const Measure As String = "Fuel Storage"
Set MeasureBox = FuelStorage
Call Assign(CheckBox3, MeasureBox, Measure)
End Sub
Private Sub CheckBox4_Click()
Const Measure As String = "Fuel Resupply"
Set MeasureBox = FuelResupply
Call Assign(CheckBox4, MeasureBox, Measure)
End Sub
Private Sub CheckBox5_Click()
Const Measure As String = "Diplomatic Clearance"
Set MeasureBox = DiplomaticClearance
Call Assign(CheckBox5, MeasureBox, Measure)
End Sub
Private Sub CheckBox6_Click()
Const Measure As String = "Force Protection"
Set MeasureBox = ForceProtection
Call Assign(CheckBox6, MeasureBox, Measure)
End Sub
Private Sub CheckBox7_Click()
Const Measure As String = "Dept of State"
Set MeasureBox = DeptOfState
Call Assign(CheckBox7, MeasureBox, Measure)
End Sub
Private Sub CheckBox8_Click()
Const Measure As String = "Military Cooperation"
Set MeasureBox = MilitaryCooperation
Call Assign(CheckBox8, MeasureBox, Measure)
End Sub
Private Sub CheckBox9_Click()
Const Measure As String = "Seaport"
Set MeasureBox = Seaport

```

```

Call Assign(CheckBox9, MeasureBox, Measure)
End Sub
'=====
=====
Private Sub Comm_Change()
Call CheckCat(Comm, CommStr)
End Sub
AirfieldInput - 6
Private Sub CommercialAirport_Change()
Call CheckCont(CommercialAirport)
End Sub
Private Sub Definitions_Click()
ShowDefinitions
End Sub
Private Sub DeptOfState_Change()
Call CheckCat(DeptOfState, DepStateStr)
End Sub
Private Sub Dining_Change()
Call CheckCat(Dining, DiningStr)
End Sub
Private Sub DiplomaticClearance_Change()
Call CheckCat(DiplomaticClearance, DipStr)
End Sub
Private Sub ForceProtection_Change()
Call CheckCat(ForceProtection, FPStr)
End Sub
Private Sub MOG_Change()
Call CheckCont(MOG)
End Sub
Private Sub Index_Click()
End Sub
Private Sub LatDeg_AfterUpdate()
If LatDeg.Value < 0 Or LatDeg.Value > 75 Then
MsgBox "Value must be between 0 and 75", vbOKOnly, "Invalid Input"
LatDeg.Value = 0
LatDeg.SetFocus
End If
End Sub
Private Sub LatMin_AfterUpdate()
If LatMin.Value < 0 Or LatMin.Value >= 60 Then
MsgBox "Value must be between 0.00 and 59.99", vbOKOnly, "Invalid Input"
LatMin.Value = 0
LatMin.SetFocus
End If
End Sub
Private Sub Lodging_Change()
Call CheckCat(Lodging, LodgingStr)
End Sub
Private Sub LongDeg_AfterUpdate()
If LongDeg.Value < 0 Or LongDeg.Value > 180 Then
MsgBox "Value must be between 0 and 180", vbOKOnly, "Invalid Input"
LongDeg.Value = 0
LongDeg.SetFocus
End If
End Sub
Private Sub LongMin_AfterUpdate()
If LongMin.Value < 0 Or LongMin.Value >= 60 Then
MsgBox "Value must be between 0.00 and 59.99", vbOKOnly, "Invalid Input"
LongMin.Value = 0
LongMin.SetFocus
End If
End Sub
Private Sub Medical_Change()

```

```

Call CheckCat(Medical, MedStr)
AirfieldInput = 7
End Sub
Private Sub MilitaryCooperation_Change()
Call CheckCat(MilitaryCooperation, MilCoopStr)
End Sub
Private Sub Mountainous_Change()
Call CheckCat(Mountainous, MountStr)
End Sub
Private Sub FuelStorage_Change()
Call CheckCont(FuelStorage)
End Sub
Private Sub Power_Change()
Call CheckCat(Power, PowerStr)
End Sub
Private Sub Railroad_Change()
Call CheckCont(Railroad)
End Sub
Private Sub RoadSystems_Change()
Call CheckCont(RoadSystems)
End Sub
Private Sub Seaport_Change()
Call CheckCont(Seaport)
End Sub
Private Sub Sewer_Change()
Call CheckCat(Sewer, SewerStr)
End Sub
Private Sub Temperature_Change()
Call CheckCat(Temperature, TempStr)
End Sub
Private Sub Terrain_Change()
Call CheckCat(Terrain, TerrainStr)
End Sub
Private Sub UrbanAreas_Change()
Call CheckCat(UrbanAreas, UrbanStr)
End Sub
Private Sub UserForm_Initialize()
Dim CatTipText As String
' Initialize an Airfield Name and format
' NextAirfield = Application.WorksheetFunction.CountA(Range("A2:A65356")) + 1
' Airfield.Value = "Airfield " &
Application.WorksheetFunction.Text(NextAirfield, 0) & _
", Country, (ICAO)"
' Write the name to the Uncertainty Data TempRow to capture uncertainty data
Worksheets("Uncertainty Data").Unprotect
Worksheets("Uncertainty Data").Cells(TempRow, 1) = Airfield.Value
Worksheets("Uncertainty Data").Protect
' Initialize Lat/Long for N or S
NorS.AddItem "N"
NorS.AddItem "S"
EorW.AddItem "E"
EorW.AddItem "W"
NorS.ListIndex = 0 ' North
EorW.ListIndex = 0 ' East
' Initialize Control Tip Text for Continuous VF Ranges
Call InitContInput("MOG", MOG)
Call InitContInput("Alternate Airfields", AlternateAirfields)
Call InitContInput("Fuel Storage", FuelStorage)
Call InitContInput("Fuel Resupply", FuelResupply)
Call InitContInput("Seaport", Seaport)
Call InitContInput("Railroad", Railroad)
Call InitContInput("Road System", RoadSystems)
Call InitContInput("Commercial Airport", CommercialAirport)
' Initialize Combo Box for Categorical VFs

```

```

Call InitComboBox("Diplomatic Clearance", DiplomaticClearance)
Call InitComboBox("Force Protection", ForceProtection)
AirfieldInput - 8
Call InitComboBox("Dept of State", DeptOfState)
Call InitComboBox("Military Cooperation", MilitaryCooperation)
Call InitComboBox("Lodging", Lodging)
Call InitComboBox("Dining", Dining)
Call InitComboBox("Medical", Medical)
Call InitComboBox("Communications", Comm)
Call InitComboBox("Power", Power)
Call InitComboBox("Potable Water", Water)
Call InitComboBox("Sewer", Sewer)
Call InitComboBox("Mountainous", Mountainous)
Call InitComboBox("Altitude", Altitude)
Call InitComboBox("Weather", Weather)
Call InitComboBox("Temperature", Temperature)
Call InitComboBox("Urban Areas", UrbanAreas)
Call InitComboBox("Terrain", Terrain)
End Sub
Private Sub InitContInput(MeasureSheetName, MeasureBox)
' Writes the Control Tip Text message using the actual range from a continuous
value function
Dim ContTipText As String
lo = 0
hi = 0
Call GetContVFRRange(MeasureSheetName, lo, hi)
ContTipText = "Value scoring range is between " & lo & " and " & hi
MeasureBox.ControlTipText = ContTipText
MeasureBox.Value = Round(lo, 0)
End Sub
Private Sub InitComboBox(MeasureName, MeasureBox)
Set MeasureSheet = ThisWorkbook.Sheets(MeasureName)
CatCount = Application.WorksheetFunction.CountA(MeasureSheet.Range("L3:L12"))
For Row = 1 To CatCount
MeasureBox.AddItem MeasureSheet.Cells(Row + 2, 12)
Next Row
MeasureBox.ListIndex = 0
End Sub
Private Sub Water_Change()
Call CheckCat(Water, WaterStr)
End Sub
Private Sub Weather_Change()
Call CheckCat(Weather, WxStr)
End Sub
Sub LoadCategories(Measure As String)
' Loads the Categories from a Categorical Value Function before displaying the
EmpiricalDist userform
Worksheets("Uncertainty Data").Unprotect
j = 1
While Worksheets("Uncertainty Data").Cells(1, j) <> Measure
j = j + 1
Wend
For i = 0 To 9
EmpiricalDist.Spreadsheet1.Cells(i + 2, 1) = Worksheets("Uncertainty
Data").Cells(2, j + i)
EmpiricalDist.Spreadsheet1.Cells(i + 2, 2) = Worksheets("Uncertainty
Data").Cells(TempRow, j +
i)
Next i
Worksheets("Uncertainty Data").Protect
End Sub
Sub LoadCont(Measure)
' Loads the values for a continuous measure triangle distribution
Set USheet = Worksheets("Uncertainty Data")

```

```

USheet.Unprotect
j = 1
AirfieldInput = 9
While USheet.Cells(1, j) <> Measure
j = j + 1
Wend
TriangleDist.Min.Value = Round(USheet.Cells(TempRow, j), 0)
TriangleDist.Mode.Value = Round(USheet.Cells(TempRow, j + 1), 0)
TriangleDist.Max.Value = Round(USheet.Cells(TempRow, j + 2), 0)
USheet.Protect
End Sub
Sub Assign(CheckBox, MeasureBox, Measure)
' If a checkbox is clicked to assign a probability distribution, disable input
If CheckBox.Value = True Then
MeasureBox.Enabled = False
MeasureBox.BackColor = &H8000000F
Else
MeasureBox.Enabled = True
MeasureBox.BackColor = &HFFFFFF
' If IsContVF(Measure) Then
' Call InitContInput(Measure, MeasureBox)
'End If
End If
End Sub
Sub ShowDistForm(Measure)
' If the ? Command Button is clicked, show the appropriate Distribution Form
Set USheet = Worksheets("Uncertainty Data")
If IsContVF(Measure) Then
Load TriangleDist
TriangleDist.Caption = "Define Probability for " & Measure & " Measure"
TriangleDist.Label2 = Measure
Call LoadCont(Measure)
TriangleDist.Show
Else
Load EmpiricalDist
EmpiricalDist.Caption = "Define Probability for " & Measure & " Measure"
EmpiricalDist.Label2 = Measure
LoadCategories(Measure)
EmpiricalDist.Show
End If
End Sub
Sub CheckCont(MeasureBox)
If Not (ValidContData(MeasureBox.Value)) Then
MsgBox "Continuous value measures must be numerical.", vbOKOnly, "Invalid Input"
MeasureBox.SetFocus
End If
End Sub
Sub CheckCat(MeasureBox, ReturnStr)
If MeasureBox.ListIndex <> -1 Then
ReturnStr = MeasureBox.Value
Else:
MsgBox "Must select a value from the drop down menu" & vbCrLf & "This value must be changed for this form to be functional again", vbOKOnly, "Invalid Input"
MeasureBox.SetFocus
End If
End Sub
Private Sub FuelResupply_Change()
Call CheckCont(FuelResupply)
End Sub
Sub WriteDataCont(NextRow, CheckBox, MeasureBox, MeasureIndex)
Set AD = Worksheets("Airfields Data")

```

```

If CheckBox.Value = False Then
If MeasureBox.Value = "" Or MeasureBox.Value = "." Then
AD.Cells(NextRow, DMCol + MeasureIndex) = 0
Else
AD.Cells(NextRow, DMCol + MeasureIndex) = MeasureBox.Value
End If
Else
AD.Cells(NextRow, DMCol + MeasureIndex) = "?"
End If
AirfieldInput = 10
End Sub

Sub WriteDataCat(NextRow, CheckBox, MeasureStr, MeasureIndex)
Set AD = Worksheets("Airfields Data")
If CheckBox.Value = False Then
AD.Cells(NextRow, DMCol + MeasureIndex) = MeasureStr
Else
AD.Cells(NextRow, DMCol + MeasureIndex) = "?"
End If
End Sub

AirfieldsHelp = 1
Option Explicit
Dim TopicCount As Integer
Dim CurrentTopic As Integer
Dim HelpSheet As Worksheet
Const HelpSheetName As String = "Airfields Help"
Const HelpFormCaption As String = "Airfield Alternatives Info"
Private Sub UserForm_Initialize()
' Executed before the form is shown
Dim Row As Integer
Set HelpSheet = ThisWorkbook.Sheets(HelpSheetName)
TopicCount = Application.WorksheetFunction.CountA(HelpSheet.Range("A:A"))
For Row = 1 To TopicCount
ComboBoxTopics.AddItem HelpSheet.Cells(Row, 1)
Next Row
ComboBoxTopics.ListIndex = 0
CurrentTopic = 1
UpdateForm
End Sub
Private Sub UpdateForm()
ComboBoxTopics.ListIndex = CurrentTopic - 1
Me.Caption = HelpFormCaption &
" (" & CurrentTopic & " of " & TopicCount & ")"
With LabelText
.Caption = HelpSheet.Cells(CurrentTopic, 2)
.AutoSize = False
.Width = 212
.AutoSize = True
End With
With Frame1
.ScrollHeight = LabelText.Height + 5
.ScrollTop = 1
End With
If CurrentTopic = 1 Then
NextButton.Enabled = True
NextButton.SetFocus
ElseIf CurrentTopic = TopicCount Then
PreviousButton.Enabled = True
PreviousButton.SetFocus
End If
PreviousButton.Enabled = CurrentTopic <> 1
NextButton.Enabled = CurrentTopic <> TopicCount
End Sub
Private Sub ComboBoxTopics_Click()

```

```

' Executed when the ComboBox is changed
CurrentTopic = ComboBoxTopics.ListIndex + 1
UpdateForm
End Sub
Private Sub PreviousButton_Click()
' Executed when the PreviousButton is clicked
If CurrentTopic <> 1 Then
CurrentTopic = CurrentTopic - 1
UpdateForm
End If
End Sub
Private Sub NextButton_Click()
' Executed when the NextButton is clicked
If CurrentTopic <> TopicCount Then
CurrentTopic = CurrentTopic + 1
UpdateForm
End If
End Sub
Private Sub ExitButton_Click()
' Executed when the ExitButton is clicked
Unload Me
AirfieldsHelp = 2
End Sub

AllWeightsHelp = 1
Private Sub CommandButton1_Click()
Unload Me
End Sub

Base = 1
Dim Weights(2), OldWeights(2)
Dim i As Byte
Private Sub CommandButton1_Click()
HowToSwing.Show
End Sub
Private Sub CommandButton2_Click()
Total = 0
' Swing the Weights
For i = 1 To 2
Total = Total + Weights(i)
Next i
For i = 1 To 2
Weights(i) = Weights(i) / Total
Next i
TextBox1.Value = Round(Weights(1), 4)
TextBox2.Value = Round(Weights(2), 4)
With Sheet3
.Range("F21") = Weights(1)
.Range("G21") = Weights(2)
End With
Label7.Caption = Round(Sheet3.Range("F22"), 4)
Label8.Caption = Round(Sheet3.Range("G22"), 4)
End Sub
Private Sub CommandButton3_Click()
Dim Valid As Boolean
Valid = True
Total = 0
For i = 1 To 2
Valid = Valid And (Weights(i) <> -1)
Total = Total + Weights(i)
Next i
If Not (Valid) Or Not (EqualOne(Total)) Then
MsgBox "Please check to ensure all weights are numerical and sum to one." &
vbCrLf & "This may require 'swinging' the weights.", vbOKOnly, "Invalid Input"

```

```

Else:
With Sheet3
    .Range("F21") = Weights(1)
    .Range("G21") = Weights(2)
End With
Unload Me
End If
End Sub
Private Sub CommandButton4_Click()
Sheet3.Range("F21") = OldWeights(1)
Sheet3.Range("G21") = OldWeights(2)
Unload Me
End Sub
Private Sub TextBox1_Change()
Weights(1) = ValidateWt(TextBox1.Value)
End Sub
Private Sub TextBox2_Change()
Weights(2) = ValidateWt(TextBox2.Value)
End Sub
Private Sub UserForm_Initialize()
' Initialize Local Weights
OldWeights(1) = Sheet3.Range("F21")
OldWeights(2) = Sheet3.Range("G21")
TextBox1.Value = Round(OldWeights(1), 4)
Base - 2
TextBox2.Value = Round(OldWeights(2), 4)
' Initialize Global Weights
Label7.Caption = Round(Sheet3.Range("F22"), 4)
Label8.Caption = Round(Sheet3.Range("G22"), 4)
For i = 1 To 2
    Weights(i) = OldWeights(i)
Next i
End Sub

BaseInfrastructure - 1
Dim Weights(3), OldWeights(3)
Dim i As Byte
Private Sub CommandButton1_Click()
HowToSwing.Show
End Sub
Private Sub CommandButton2_Click()
Total = 0
' Swing the Weights
For i = 1 To 3
    Total = Total + Weights(i)
Next i
For i = 1 To 3
    Weights(i) = Weights(i) / Total
Next i
TextBox1.Value = Round(Weights(1), 4)
TextBox2.Value = Round(Weights(2), 4)
TextBox3.Value = Round(Weights(3), 4)
With Sheet3
    .Range("F13") = Weights(1)
    .Range("G13") = Weights(2)
    .Range("H13") = Weights(3)
End With
Label7.Caption = Round(Sheet3.Range("F14"), 4)
Label8.Caption = Round(Sheet3.Range("G14"), 4)
Label9.Caption = Round(Sheet3.Range("H14"), 4)
End Sub
Private Sub CommandButton3_Click()
Dim Valid As Boolean
Valid = True

```

```

Total = 0
For i = 1 To 3
Valid = Valid And (Weights(i) <> -1)
Total = Total + Weights(i)
Next i
If (Not (Valid)) Or Not (EqualOne(Total)) Then
MsgBox "Please check to ensure all weights are numerical and sum to one." &
vbCrLf & "This may require 'swinging' the weights.", vbOKOnly, "Invalid Input"
Else:
With Sheet3
.Range("F13") = Weights(1)
.Range("G13") = Weights(2)
.Range("H13") = Weights(3)
End With
Unload Me
End If
End Sub
Private Sub CommandButton4_Click()
Sheet3.Range("F13") = OldWeights(1)
Sheet3.Range("G13") = OldWeights(2)
Sheet3.Range("H13") = OldWeights(3)
Unload Me
End Sub
Private Sub Label5_Click()
End Sub
Private Sub TextBox1_Change()
Weights(1) = ValidateWt(TextBox1.Value)
End Sub
Private Sub TextBox2_Change()
BaseInfrastructure = 2
Weights(2) = ValidateWt(TextBox2.Value)
End Sub
Private Sub TextBox3_Change()
Weights(3) = ValidateWt(TextBox3.Value)
End Sub
Private Sub UserForm_Initialize()
' Initialize Local Weights
OldWeights(1) = Sheet3.Range("F13")
OldWeights(2) = Sheet3.Range("G13")
OldWeights(3) = Sheet3.Range("H13")
TextBox1.Value = Round(OldWeights(1), 4)
TextBox2.Value = Round(OldWeights(2), 4)
TextBox3.Value = Round(OldWeights(3), 4)
' Initialize Global Weights
Label7.Caption = Round(Sheet3.Range("F14"), 4)
Label8.Caption = Round(Sheet3.Range("G14"), 4)
Label9.Caption = Round(Sheet3.Range("H14"), 4)
For i = 1 To 3
Weights(i) = OldWeights(i)
Next i
End Sub

Definitions = 1
Option Explicit
Dim TopicCount As Integer
Dim CurrentTopic As Integer
Dim HelpSheet As Worksheet
Dim CurrentSheet As String
Const HelpSheetName As String = "Definitions"
Const HelpFormCaption As String = "Definitions of Value Measures"
Private Sub UserForm_Initialize()
' Executed before the form is shown
Dim Row As Integer
Set HelpSheet = ThisWorkbook.Sheets(HelpSheetName)

```

```

TopicCount = Application.WorksheetFunction.CountA(HelpSheet.Range("A:A"))
For Row = 1 To TopicCount
    ComboBoxTopics.AddItem HelpSheet.Cells(Row, 1)
Next Row
CurrentSheet = ActiveSheet.Name
'MsgBox CurrentSheet
CurrentTopic = 1
Select Case CurrentSheet
Case "Critical Leg"
    CurrentTopic = 2
Case "Delta Flight Length"
    CurrentTopic = 3
Case "Alternate Airfields"
    CurrentTopic = 4
Case "Fuel Storage"
    CurrentTopic = 5
Case "Fuel Resupply"
    CurrentTopic = 6
Case "Diplomatic Clearance"
    CurrentTopic = 7
Case "Force Protection"
    CurrentTopic = 8
Case "Dept of State"
    CurrentTopic = 9
Case "Military Cooperation"
    CurrentTopic = 10
Case "Seaport"
    CurrentTopic = 11
Case "Railroad"
    CurrentTopic = 12
Case "Road System"
    CurrentTopic = 13
Case "Commercial Airport"
    CurrentTopic = 14
Case "Lodging"
    CurrentTopic = 15
Case "Dining"
    CurrentTopic = 16
Case "Medical"
    CurrentTopic = 17
Case "Communications"
    CurrentTopic = 18
Case "Power"
    CurrentTopic = 19
Case "Potable Water"
    CurrentTopic = 20
Case "Sewer"
    CurrentTopic = 21
Case "Mountainous"
    CurrentTopic = 22
Case "Altitude"
    CurrentTopic = 23
Case "Weather"
    CurrentTopic = 24
Case "Temperature"
    CurrentTopic = 25
Case "Urban Areas"
    CurrentTopic = 26
Definitions - 2
Case "Terrain"
    CurrentTopic = 27
Case "Probability"
    CurrentTopic = 28
Case "National Security"

```

```

CurrentTopic = 29
End Select
'MsgBox CurrentTopic
ComboBoxTopics.ListIndex = CurrentTopic - 1
UpdateForm
End Sub
Private Sub UpdateForm()
ComboBoxTopics.ListIndex = CurrentTopic - 1
Me.Caption = HelpFormCaption &
" (" & CurrentTopic & " of " & TopicCount & ")"
With LabelText
.Caption = HelpSheet.Cells(CurrentTopic, 2)
.AutoSize = False
.Width = 212
.AutoSize = True
End With
With Frame1
.ScrollHeight = LabelText.Height + 5
.ScrollTop = 1
End With
If CurrentTopic = 1 Then
NextButton.Enabled = True
NextButton.SetFocus
ElseIf CurrentTopic = TopicCount Then
PreviousButton.Enabled = True
PreviousButton.SetFocus
End If
PreviousButton.Enabled = CurrentTopic <> 1
NextButton.Enabled = CurrentTopic <> TopicCount
End Sub
Private Sub ComboBoxTopics_Click()
' Executed when the ComboBox is changed
CurrentTopic = ComboBoxTopics.ListIndex + 1
UpdateForm
End Sub
Private Sub PreviousButton_Click()
' Executed when the PreviousButton is clicked
If CurrentTopic <> 1 Then
CurrentTopic = CurrentTopic - 1
UpdateForm
End If
End Sub
Private Sub NextButton_Click()
' Executed when the NextButton is clicked
If CurrentTopic <> TopicCount Then
CurrentTopic = CurrentTopic + 1
UpdateForm
End If
End Sub
Private Sub ExitButton_Click()
' Executed when the ExitButton is clicked
Unload Me
End Sub

DeleteOrig = 1
Dim AirfieldIndex As Long
Dim Count As Integer
Private Sub AirfieldBox_Change()
AirfieldIndex = AirfieldBox.ListIndex
End Sub
Private Sub CommandButton1_Click()
If AirfieldIndex = -1 Then
MsgBox "You must select a valid airfield", vbOKOnly, "Invalid Input"
Else:

```

```

Airfield = AirfieldBox.Value
Msg = "Are you sure you want to Remove " & Airfield & " ?"
Ans = MsgBox(Msg, vbYesNo, "Remove Airfield?")
If (Ans = vbYes) Then
    ActiveSheet.Unprotect
    AirfieldRow = Application.WorksheetFunction.Text(AirfieldIndex + FirstOD, 0)
    SelectRange = "A" & AirfieldRow & ":J" & AirfieldRow
    Range(SelectRange).Select
    Selection.Delete Shift:=xlUp
    Range("A1").Select
    ActiveSheet.Protect
    Unload Me
End If
End If
End Sub
Private Sub CommandButton2_Click()
Unload Me
End Sub
Private Sub UserForm_Initialize()
Count = Application.WorksheetFunction.CountA(Range("A1:A65356"))
If Count = 1 Then
    MsgBox "Cannot delete last origin. This will delete some formula
calculations." & vbCrLf &
"Suggest editing current airfield or adding new airfield", vbOKOnly, "Must
have one or
igin"
Unload Me
End If
For Row = 1 To Count
    AirfieldBox.AddItem Cells(Row + FirstOD - 1, 1)
Next Row
AirfieldBox.ListIndex = 0
AirfieldIndex = 0
End Sub

EmpiricalDist = 1
Private Sub CommandButton1_Click()
Measure = Label2
Set USheet = Worksheets("Uncertainty Data")
Set Sh = Spreadsheet1
Count =
Application.WorksheetFunction.CountA(Worksheets(Measure).Range("L3:L12"))
If Sh.Cells(12, 2) < 0.998 Or Sh.Cells(12, 2) > 1.002 Then
    MsgBox "The sum of all the probabilities must equal one.", vbOKOnly, "Invalid
Input"
Else
    ExtraVal = False
    For i = Count + 2 To 11
        If (Sh.Cells(i, 2) <> 0) Then
            ExtraVal = True
        End If
    Next i
    If ExtraVal Then
        MsgBox "You have input a probability value for a category that does not
exist", vbOKOnly,
        "Invalid Input"
        ExtraVal = False
    Else
        NotANum = False
        For i = 2 To Count + 1
            Sheets("Weights").Range("B101") = Sh.Cells(i, 2)
            IsANumber = Sheets("Weights").Range("C101")
            If Not (IsANumber) Then
                NotANum = True
            End If
        Next i
        If NotANum Then
            MsgBox "All categories must have a probability value.", vbOKOnly,
            "Invalid Input"
        End If
    End If
End Sub

```

```

End If
Next i
If NotANum Then
MsgBox "One of your probability inputs is not a number from 0 to 1", vbOKOnly,
"Invalid Input"
NotANum = False 'Reset to allow for correction
Else 'Otherwise update uncertainty info in TempRow
USheet.Unprotect
i = UCol
While USheet.Cells(1, i) <> Measure
i = i + 1
Wend
For j = 0 To 9
USheet.Cells(TempRow, i + j) = Sh.Cells(j + 2, 2)
Next j
USheet.Protect
Unload Me
End If
End If
End If
End Sub
Private Sub CommandButton2_Click()
Unload Me
End Sub
Private Sub UserForm_Click()
End Sub

EnrouteDiversion - 1
Dim Weights(3), OldWeights(3)
Dim i As Byte
Private Sub CommandButton1_Click()
HowToSwing.Show
End Sub
Private Sub CommandButton2_Click()
Total = 0
' Swing the Weights
For i = 1 To 3
Total = Total + Weights(i)
Next i
For i = 1 To 3
Weights(i) = Weights(i) / Total
Next i
TextBox1.Value = Round(Weights(1), 4)
TextBox2.Value = Round(Weights(2), 4)
TextBox3.Value = Round(Weights(3), 4)
With Sheet3
.Range("B9") = Weights(1)
.Range("C9") = Weights(2)
.Range("D9") = Weights(3)
End With
Label7.Caption = Round(Sheet3.Range("B10"), 4)
Label8.Caption = Round(Sheet3.Range("C10"), 4)
Label9.Caption = Round(Sheet3.Range("D10"), 4)
End Sub
Private Sub CommandButton3_Click()
Dim Valid As Boolean
Valid = True
Total = 0
For i = 1 To 3
Valid = Valid And (Weights(i) <> -1)
Total = Total + Weights(i)
Next i
If (Not (Valid)) Or Not (EqualOne(Total)) Then

```

```

MsgBox "Please check to ensure all weights are numerical and sum to one." &
vbCrLf & "This may require 'swinging' the weights.", vbOKOnly, "Invalid Input"
Else:
With Sheet3
    .Range("B9") = Weights(1)
    .Range("C9") = Weights(2)
    .Range("D9") = Weights(3)
End With
Unload Me
End If
End Sub
Private Sub CommandButton4_Click()
Sheet3.Range("B9") = OldWeights(1)
Sheet3.Range("C9") = OldWeights(2)
Sheet3.Range("D9") = OldWeights(3)
Unload Me
End Sub
Private Sub TextBox1_Change()
Weights(1) = ValidateWt(TextBox1.Value)
End Sub
Private Sub TextBox2_Change()
Weights(2) = ValidateWt(TextBox2.Value)
End Sub
Private Sub TextBox3_Change()
Weights(3) = ValidateWt(TextBox3.Value)
End Sub
EnrouteDiversion - 2
Private Sub UserForm_Initialize()
    ' Initialize Local Weights
    OldWeights(1) = Sheet3.Range("B9")
    OldWeights(2) = Sheet3.Range("C9")
    OldWeights(3) = Sheet3.Range("D9")
    TextBox1.Value = Round(OldWeights(1), 4)
    TextBox2.Value = Round(OldWeights(2), 4)
    TextBox3.Value = Round(OldWeights(3), 4)
    ' Initialize Global Weights
    Label7.Caption = Round(Sheet3.Range("B10"), 4)
    Label8.Caption = Round(Sheet3.Range("C10"), 4)
    Label9.Caption = Round(Sheet3.Range("D10"), 4)
    For i = 1 To 3
        Weights(i) = OldWeights(i)
    Next i
End Sub

EnrouteScore - 1
Dim Weights(2), OldWeights(2)
Dim i As Byte
Private Sub CommandButton1_Click()
HowToSwing.Show
End Sub
Private Sub CommandButton2_Click()
Total = 0
    ' Swing the Weights
    For i = 1 To 2
        Total = Total + Weights(i)
    Next i
    For i = 1 To 2
        Weights(i) = Weights(i) / Total
    Next i
    TextBox1.Value = Round(Weights(1), 4)
    TextBox2.Value = Round(Weights(2), 4)
    With Sheet3
        .Range("B5") = Weights(1)
        .Range("C5") = Weights(2)
    End With
End Sub

```

```

End With
Label7.Caption = Round(Sheet3.Range("B6"), 4)
Label8.Caption = Round(Sheet3.Range("C6"), 4)
End Sub
Private Sub CommandButton3_Click()
Dim Valid As Boolean
Valid = True
Total = 0
For i = 1 To 2
Valid = Valid And (Weights(i) <> -1)
Total = Total + Weights(i)
Next i
If (Not (Valid)) Or Not (EqualOne(Total)) Then
MsgBox "Please check to ensure all weights are numerical and sum to one." &
vbCrLf & "This may require 'swinging' the weights.", vbOKOnly, "Invalid Input"
Else:
With Sheet3
.Range("B5") = Weights(1)
.Range("C5") = Weights(2)
End With
Unload Me
End If
End Sub
Private Sub CommandButton4_Click()
Sheet3.Range("B5") = OldWeights(1)
Sheet3.Range("C5") = OldWeights(2)
Unload Me
End Sub
Private Sub TextBox1_Change()
Weights(1) = ValidateWt(TextBox1.Value)
End Sub
Private Sub TextBox2_Change()
Weights(2) = ValidateWt(TextBox2.Value)
End Sub
Private Sub UserForm_Initialize()
' Initialize Local Weights
OldWeights(1) = Sheet3.Range("B5")
OldWeights(2) = Sheet3.Range("C5")
EnrouteScore = 2
TextBox1.Value = Round(OldWeights(1), 4)
TextBox2.Value = Round(OldWeights(2), 4)
' Initialize Global Weights
Label7.Caption = Round(Sheet3.Range("B6"), 4)
Label8.Caption = Round(Sheet3.Range("C6"), 4)
For i = 1 To 2
Weights(i) = OldWeights(i)
Next i
End Sub

Facilities = 1
Dim Weights(3), OldWeights(3)
Dim i As Byte
Private Sub CommandButton1_Click()
HowToSwing.Show
End Sub
Private Sub CommandButton2_Click()
Total = 0
' Swing the Weights
For i = 1 To 3
Total = Total + Weights(i)
Next i
For i = 1 To 3
Weights(i) = Weights(i) / Total
Next i

```

```

TextBox1.Value = Round(Weights(1), 4)
TextBox2.Value = Round(Weights(2), 4)
TextBox3.Value = Round(Weights(3), 4)
With Sheet3
    .Range("B17") = Weights(1)
    .Range("C17") = Weights(2)
    .Range("D17") = Weights(3)
End With
Label7.Caption = Round(Sheet3.Range("B18"), 4)
Label8.Caption = Round(Sheet3.Range("C18"), 4)
Label9.Caption = Round(Sheet3.Range("D18"), 4)
End Sub
Private Sub CommandButton3_Click()
Dim Valid As Boolean
Valid = True
Total = 0
For i = 1 To 3
    Valid = Valid And (Weights(i) <> -1)
    Total = Total + Weights(i)
Next i
If (Not (Valid)) Or Not (EqualOne(Total)) Then
    MsgBox "Please check to ensure all weights are numerical and sum to one." &
        vbCrLf & "This may require 'swinging' the weights.", vbOKOnly, "Invalid Input"
Else:
    With Sheet3
        .Range("B17") = Weights(1)
        .Range("C17") = Weights(2)
        .Range("D17") = Weights(3)
    End With
    Unload Me
End If
End Sub
Private Sub CommandButton4_Click()
Sheet3.Range("B17") = OldWeights(1)
Sheet3.Range("C17") = OldWeights(2)
Sheet3.Range("D17") = OldWeights(3)
Unload Me
End Sub
Private Sub TextBox1_Change()
Weights(1) = ValidateWt(TextBox1.Value)
End Sub
Private Sub TextBox2_Change()
Weights(2) = ValidateWt(TextBox2.Value)
End Sub
Private Sub TextBox3_Change()
Weights(3) = ValidateWt(TextBox3.Value)
End Sub
Facilities - 2
Private Sub UserForm_Initialize()
    ' Initialize Local Weights
    OldWeights(1) = Sheet3.Range("B17")
    OldWeights(2) = Sheet3.Range("C17")
    OldWeights(3) = Sheet3.Range("D17")
    TextBox1.Value = Round(OldWeights(1), 4)
    TextBox2.Value = Round(OldWeights(2), 4)
    TextBox3.Value = Round(OldWeights(3), 4)
    ' Initialize Global Weights
    Label7.Caption = Round(Sheet3.Range("B18"), 4)
    Label8.Caption = Round(Sheet3.Range("C18"), 4)
    Label9.Caption = Round(Sheet3.Range("D18"), 4)
    For i = 1 To 3
        Weights(i) = OldWeights(i)
    Next i
End Sub

```

```

Fuel - 1
Dim Weights(2), OldWeights(2)
Dim i As Byte
Private Sub CommandButton1_Click()
HowToSwing.Show
End Sub
Private Sub CommandButton2_Click()
Total = 0
' Swing the Weights
For i = 1 To 2
Total = Total + Weights(i)
Next i
For i = 1 To 2
Weights(i) = Weights(i) / Total
Next i
TextBox1.Value = Round(Weights(1), 4)
TextBox2.Value = Round(Weights(2), 4)
With Sheet3
.Range("E9") = Weights(1)
.Range("F9") = Weights(2)
End With
Label7.Caption = Round(Sheet3.Range("E10"), 4)
Label8.Caption = Round(Sheet3.Range("F10"), 4)
End Sub
Private Sub CommandButton3_Click()
Dim Valid As Boolean
Valid = True
Total = 0
For i = 1 To 2
Valid = Valid And (Weights(i) <> -1)
Total = Total + Weights(i)
Next i
If (Not (Valid)) Or Not (EqualOne(Total)) Then
MsgBox "Please check to ensure all weights are numerical and sum to one." &
vbCrLf & "This may require 'swinging' the weights.", vbOKOnly, "Invalid Input"
Else:
With Sheet3
.Range("E9") = Weights(1)
.Range("F9") = Weights(2)
End With
Unload Me
End If
End Sub
Private Sub CommandButton4_Click()
Sheet3.Range("E9") = OldWeights(1)
Sheet3.Range("F9") = OldWeights(2)
Unload Me
End Sub
Private Sub TextBox1_Change()
Weights(1) = ValidateWt(TextBox1.Value)
End Sub
Private Sub TextBox2_Change()
Weights(2) = ValidateWt(TextBox2.Value)
End Sub
Private Sub UserForm_Initialize()
' Initialize Local Weights
OldWeights(1) = Sheet3.Range("E9")
OldWeights(2) = Sheet3.Range("F9")
Fuel - 2
TextBox1.Value = Round(OldWeights(1), 4)
TextBox2.Value = Round(OldWeights(2), 4)
' Initialize Global Weights
Label7.Caption = Round(Sheet3.Range("E10"), 4)

```

```

Label8.Caption = Round(Sheet3.Range("F10"), 4)
For i = 1 To 2
Weights(i) = OldWeights(i)
Next i
End Sub

Geography - 1
Dim Weights(2), OldWeights(2)
Dim i As Byte
Private Sub CommandButton1_Click()
HowToSwing.Show
End Sub
Private Sub CommandButton2_Click()
Total = 0
' Swing the Weights
For i = 1 To 2
Total = Total + Weights(i)
Next i
For i = 1 To 2
Weights(i) = Weights(i) / Total
Next i
TextBox1.Value = Round(Weights(1), 4)
TextBox2.Value = Round(Weights(2), 4)
With Sheet3
.Range("I13") = Weights(1)
.Range("J13") = Weights(2)
End With
Label7.Caption = Round(Sheet3.Range("I14"), 4)
Label8.Caption = Round(Sheet3.Range("J14"), 4)
End Sub
Private Sub CommandButton3_Click()
Dim Valid As Boolean
Valid = True
Total = 0
For i = 1 To 2
Valid = Valid And (Weights(i) <> -1)
Total = Total + Weights(i)
Next i
If (Not (Valid)) Or Not (EqualOne(Total)) Then
MsgBox "Please check to ensure all weights are numerical and sum to one." &
vbCrLf & "This may require 'swinging' the weights.", vbOKOnly, "Invalid Input"
Else:
With Sheet3
.Range("I13") = Weights(1)
.Range("J13") = Weights(2)
End With
Unload Me
End If
End Sub
Private Sub CommandButton4_Click()
Sheet3.Range("I13") = OldWeights(1)
Sheet3.Range("J13") = OldWeights(2)
Unload Me
End Sub
Private Sub Label8_Click()
End Sub
Private Sub TextBox1_Change()
Weights(1) = ValidateWt(TextBox1.Value)
End Sub
Private Sub TextBox2_Change()
Weights(2) = ValidateWt(TextBox2.Value)
End Sub
Private Sub UserForm_Initialize()
Geography - 2

```

```

' Initialize Local Weights
OldWeights(1) = Sheet3.Range("I13")
OldWeights(2) = Sheet3.Range("J13")
TextBox1.Value = Round(OldWeights(1), 4)
TextBox2.Value = Round(OldWeights(2), 4)
' Initialize Global Weights
Label7.Caption = Round(Sheet3.Range("I14"), 4)
Label8.Caption = Round(Sheet3.Range("J14"), 4)
For i = 1 To 2
Weights(i) = OldWeights(i)
Next i
End Sub

GroundTransportation - 1
Dim Weights(4), OldWeights(4)
Dim i As Byte
Private Sub CommandButton1_Click()
HowToSwing.Show
End Sub
Private Sub CommandButton2_Click()
Total = 0
' Swing the Weights
For i = 1 To 4
Total = Total + Weights(i)
Next i
For i = 1 To 4
Weights(i) = Weights(i) / Total
Next i
TextBox1.Value = Round(Weights(1), 4)
TextBox6.Value = Round(Weights(2), 4)
TextBox5.Value = Round(Weights(3), 4)
TextBox4.Value = Round(Weights(4), 4)
With Sheet3
.Range("B13") = Weights(1)
.Range("C13") = Weights(2)
.Range("D13") = Weights(3)
.Range("E13") = Weights(4)
End With
Label8.Caption = Round(Sheet3.Range("B14"), 4) 'Seaport cell B22
Label9.Caption = Round(Sheet3.Range("C14"), 4) 'Railroad cell C22
Label10.Caption = Round(Sheet3.Range("D14"), 4) 'Road cell D22
Label11.Caption = Round(Sheet3.Range("E14"), 4) 'Comm Airport cell E22
End Sub
Private Sub CommandButton3_Click()
Dim Valid As Boolean
Valid = True
Total = 0
For i = 1 To 4
Valid = Valid And (Weights(i) <> -1)
Total = Total + Weights(i)
Next i
If (Not (Valid)) Or Not (EqualOne(Total)) Then
MsgBox "Please check to ensure all weights are numerical and sum to one." &
vbCrLf & "This may require 'swinging' the weights.", vbOKOnly, "Invalid Input"
Else:
With Sheet3
.Range("B13") = Weights(1)
.Range("C13") = Weights(2)
.Range("D13") = Weights(3)
.Range("E13") = Weights(4)
End With
Unload Me
End If
End Sub

```

```

Private Sub CommandButton4_Click()
Sheet3.Range("B13") = OldWeights(1)
Sheet3.Range("C13") = OldWeights(2)
Sheet3.Range("D13") = OldWeights(3)
Sheet3.Range("E13") = OldWeights(4)
Unload Me
End Sub
Private Sub Label11_Click()
End Sub
Private Sub TextBox1_Change()
GroundTransportation - 2
Weights(1) = ValidateWt(TextBox1.Value)
End Sub
Private Sub TextBox6_Change()
Weights(2) = ValidateWt(TextBox6.Value)
End Sub
Private Sub TextBox5_Change()
Weights(3) = ValidateWt(TextBox5.Value)
End Sub
Private Sub TextBox4_Change()
Weights(4) = ValidateWt(TextBox4.Value)
End Sub
Private Sub UserForm_Initialize()
' Initialize Local Weights
OldWeights(1) = Sheet3.Range("B13")
OldWeights(2) = Sheet3.Range("C13")
OldWeights(3) = Sheet3.Range("D13")
OldWeights(4) = Sheet3.Range("E13")
TextBox1.Value = Round(OldWeights(1), 4)
TextBox6.Value = Round(OldWeights(2), 4)
TextBox5.Value = Round(OldWeights(3), 4)
TextBox4.Value = Round(OldWeights(4), 4)
' Initialize Global Weights
Label8.Caption = Round(Sheet3.Range("B14"), 4)
Label9.Caption = Round(Sheet3.Range("C14"), 4)
Label10.Caption = Round(Sheet3.Range("D14"), 4)
Label11.Caption = Round(Sheet3.Range("E14"), 4)
For i = 1 To 4
Weights(i) = OldWeights(i)
Next i
End Sub

HierarchyHelp - 1
Private Sub CommandButton1_Click()
Unload Me
End Sub

HostNationRelations - 1
Dim Weights(4), OldWeights(4)
Dim i As Byte
Private Sub CommandButton1_Click()
HowToSwing.Show
End Sub
Private Sub CommandButton2_Click()
Total = 0
' Swing the Weights
For i = 1 To 4
Total = Total + Weights(i)
Next i
For i = 1 To 4
Weights(i) = Weights(i) / Total
Next i
TextBox1.Value = Round(Weights(1), 4)
TextBox2.Value = Round(Weights(2), 4)

```

```

TextBox3.Value = Round(Weights(3), 4)
TextBox4.Value = Round(Weights(4), 4)
With Sheet3
    .Range("G9") = Weights(1)
    .Range("H9") = Weights(2)
    .Range("I9") = Weights(3)
    .Range("J9") = Weights(4)
End With
Label8.Caption = Round(Sheet3.Range("G10"), 4)
Label9.Caption = Round(Sheet3.Range("H10"), 4)
Label10.Caption = Round(Sheet3.Range("I10"), 4)
Label11.Caption = Round(Sheet3.Range("J10"), 4)
End Sub
Private Sub CommandButton3_Click()
Dim Valid As Boolean
Valid = True
Total = 0
For i = 1 To 4
    Valid = Valid And (Weights(i) <> -1)
    Total = Total + Weights(i)
Next i
If (Not (Valid)) Or Not (EqualOne(Total)) Then
    MsgBox "Please check to ensure all weights are numerical and sum to one." &
        vbCrLf & "This may require 'swinging' the weights.", vbOKOnly, "Invalid Input"
Else:
    With Sheet3
        .Range("G9") = Weights(1)
        .Range("H9") = Weights(2)
        .Range("I9") = Weights(3)
        .Range("J9") = Weights(4)
    End With
    Unload Me
End If
End Sub
Private Sub CommandButton4_Click()
Sheet3.Range("G9") = OldWeights(1)
Sheet3.Range("H9") = OldWeights(2)
Sheet3.Range("I9") = OldWeights(3)
Sheet3.Range("J9") = OldWeights(4)
Unload Me
End Sub
Private Sub Label11_Click()
End Sub
Private Sub TextBox1_Change()
HostNationRelations = 2
Weights(1) = ValidateWt(TextBox1.Value)
End Sub
Private Sub TextBox2_Change()
Weights(2) = ValidateWt(TextBox2.Value)
End Sub
Private Sub TextBox3_Change()
Weights(3) = ValidateWt(TextBox3.Value)
End Sub
Private Sub TextBox4_Change()
Weights(4) = ValidateWt(TextBox4.Value)
End Sub
Private Sub UserForm_Initialize()
    ' Initialize Local Weights
    OldWeights(1) = Sheet3.Range("G9")
    OldWeights(2) = Sheet3.Range("H9")
    OldWeights(3) = Sheet3.Range("I9")
    OldWeights(4) = Sheet3.Range("J9")
    TextBox1.Value = Round(OldWeights(1), 4)
    TextBox2.Value = Round(OldWeights(2), 4)

```

```

TextBox3.Value = Round(OldWeights(3), 4)
TextBox4.Value = Round(OldWeights(4), 4)
' Initialize Global Weights
Label8.Caption = Round(Sheet3.Range("G10"), 4)
Label9.Caption = Round(Sheet3.Range("H10"), 4)
Label10.Caption = Round(Sheet3.Range("I10"), 4)
Label11.Caption = Round(Sheet3.Range("J10"), 4)
For i = 1 To 4
    Weights(i) = OldWeights(i)
Next i
End Sub

HowToSwing - 1
Private Sub CommandButton1_Click()
Unload Me
End Sub
Private Sub UserForm_Click()
End Sub

MainHelp - 1
Option Explicit
Dim TopicCount As Integer
Dim CurrentTopic As Integer
Dim HelpSheet As Worksheet
Const HelpSheetName As String = "Main Help"
Const HelpFormCaption As String = "Main Page Info"
Private Sub LabelText_Click()
End Sub
Private Sub UserForm_Initialize()
    ' Executed before the form is shown
    Dim Row As Integer
    Set HelpSheet = ThisWorkbook.Sheets(HelpSheetName)
    TopicCount = Application.WorksheetFunction.CountA(HelpSheet.Range("A:A"))
    For Row = 1 To TopicCount
        ComboBoxTopics.AddItem HelpSheet.Cells(Row, 1)
    Next Row
    ComboBoxTopics.ListIndex = 0
    CurrentTopic = 1
    UpdateForm
End Sub
Private Sub UpdateForm()
    ComboBoxTopics.ListIndex = CurrentTopic - 1
    Me.Caption = HelpFormCaption &
    "(" & CurrentTopic & " of " & TopicCount & ")"
    With LabelText
        .Caption = HelpSheet.Cells(CurrentTopic, 2)
        .AutoSize = False
        .Width = 212
        .AutoSize = True
    End With
    With Frame1
        .ScrollHeight = LabelText.Height + 5
        .ScrollTop = 1
    End With
    If CurrentTopic = 1 Then
        NextButton.Enabled = True
        NextButton.SetFocus
    ElseIf CurrentTopic = TopicCount Then
        PreviousButton.Enabled = True
        PreviousButton.SetFocus
    End If
    PreviousButton.Enabled = CurrentTopic <> 1
    NextButton.Enabled = CurrentTopic <> TopicCount
End Sub

```

```

Private Sub ComboBoxTopics_Click()
    ' Executed when the ComboBox is changed
    CurrentTopic = ComboBoxTopics.ListIndex + 1
    UpdateForm
End Sub

Private Sub PreviousButton_Click()
    ' Executed when the PreviousButton is clicked
    If CurrentTopic <> 1 Then
        CurrentTopic = CurrentTopic - 1
        UpdateForm
    End If
End Sub

Private Sub NextButton_Click()
    ' Executed when the NextButton is clicked
    If CurrentTopic <> TopicCount Then
        CurrentTopic = CurrentTopic + 1
        UpdateForm
    End If
End Sub

MainHelp = 2

Private Sub ExitButton_Click()
    ' Executed when the ExitButton is clicked
    Unload Me
End Sub

OpSensitivityHelp = 1

Private Sub CommandButton1_Click()
    Unload Me
End Sub

Private Sub UserForm_Click()
End Sub

RunModelHelp = 1

Private Sub CommandButton1_Click()
    Unload Me
End Sub

RunModelMenu = 1

Dim Count As Integer
Dim Current As Integer
Dim DataSheet As Worksheet
Dim Analyze As String
Dim ByTier As Byte
Dim SensitivityOption As Byte
Const DataSheetName As String = "Weights"
Const SRow = 113 ' Row where Sensitivity weights are stored

Private Sub CommandButton1_Click() 'SHOW RANKINGS
    Application.ScreenUpdating = False
    Worksheets("Rankings").Activate
    Worksheets("Rankings").Visible = True
    Worksheets("Main").Visible = False
    Call CreateRankings(ByTier)
    Unload Me
    Application.ScreenUpdating = True
End Sub

Private Sub CommandButton2_Click() 'PERFORM SENSITIVITY ANALYSIS
    Dim SensType As String
    Dim i, Branch, Tier, RefRow As Byte
    Dim Sum, Denom As Single
    Dim UpdateWt As String
    Set WeightSheet = Worksheets("Weights")
    Set DataSheet = Worksheets("Airfields Data")
    Set ValueSheet = Worksheets("Values")
    If SensitivityOption = 1 Then

```

```

SensType = "Local"
Else: SensType = "Global"
End If
Application.ScreenUpdating = False
WeightSheet.Unprotect
DataSheet.Unprotect
ValueSheet.Unprotect
WeightSheet.Rows("132:65356").Clear
'----- LOCAL PROPORTIONAL SENSITIVITY -----
-----
If SensitivityOption = 1 Then
'Delete Previous Values
WeightSheet.Range("B121:AN123").Clear
'Look up Branch number of selected measure
Branch = WeightSheet.Cells(SRow + 3, Current)
RefRow = 106 'Row to grab weighting to apply to Measure Scores
Sum = 0
i = 2
'Sum the global weights for all values in tier
' In the same loop, set weights for when Sensitivity measure is 1, the rest are
zero
While WeightSheet.Cells(SRow + 3, i) <= Branch
If WeightSheet.Cells(SRow + 3, i) = Branch Then
'Save value to restore later
WeightSheet.Cells(SRow + 10, i) = WeightSheet.Cells(SRow + 6, i)
If i = Current Then
CurrentWt = WeightSheet.Cells(SRow + 6, i)
WeightSheet.Cells(SRow + 8, i) = 0
WeightSheet.Cells(SRow + 9, i) = 1
Else
WeightSheet.Cells(SRow + 9, i) = 0
Sum = WeightSheet.Cells(SRow + 6, i) + Sum
End If
RunModelMenu - 2
End If
i = i + 1
Wend
'Find common denominator for rest of tier when Sensitivity measure is zero
Denom = Sum - WeightSheet.Cells(SRow + 8, Current)
'Determine weights for other values when Sensitivity measure is zero
i = 2
While WeightSheet.Cells(SRow + 3, i) <= Branch
If (WeightSheet.Cells(SRow + 3, i) = Branch) And (i <> Current) Then
If Denom <> 0 Then
WeightSheet.Cells(SRow + 8, i) = WeightSheet.Cells(SRow + 6, i) / Denom
Else: WeightSheet.Cells(SRow + 8, i) = 0
End If
End If
i = i + 1
Wend
'Run Value Scores for when Sensitivity measure is ZERO
' First, write the newly computed weights to the active weights spreadsheet
i = 2
While WeightSheet.Cells(SRow + 3, i) <= Branch
If WeightSheet.Cells(SRow + 3, i) = Branch Then
UpdateWt = WeightSheet.Cells(SRow + 4, i)
WeightSheet.Range(UpdateWt) = WeightSheet.Cells(SRow + 8, i)
End If
i = i + 1
Wend
Call ComputeSensitivityValues(RefRow, 0)
'Run Value Scores for when Sensitivity measure is ONE
' First, write the newly computed weights to the active weights spreadsheet
i = 2

```

```

While WeightSheet.Cells(SRow + 3, i) <= Branch
If WeightSheet.Cells(SRow + 3, i) = Branch Then
UpdateWt = WeightSheet.Cells(SRow + 4, i)
WeightSheet.Range(UpdateWt) = WeightSheet.Cells(SRow + 9, i)
End If
i = i + 1
Wend
Call ComputeSensitivityValues(RefRow, 1)
'Run Value Scores when Sensitivity measure is ONE
'Restore weights from before Sensitivity analysis
i = 2
While WeightSheet.Cells(SRow + 3, i) <= Branch
If WeightSheet.Cells(SRow + 3, i) = Branch Then
UpdateWt = WeightSheet.Cells(SRow + 4, i)
WeightSheet.Range(UpdateWt) = WeightSheet.Cells(SRow + 10, i)
End If
i = i + 1
Wend
-----GLOBAL PROPORTIONAL SENSITIVITY-----
-----
Else
'Delete Previous Values
WeightSheet.Range("B121:AN123").Clear
'Determine the current weight and find the denominator for global proportional
CurrentWt = WeightSheet.Cells(SRow + 7, Current)
Denom = 1 - CurrentWt
Tier = WeightSheet.Cells(SRow + 1, Current)
ID = WeightSheet.Cells(SRow + 2, Current)
***** Compute weights for when Current Value becomes 0
'Start from the end (or bottom) to find children of Current Value
' If a child is found, zero it out, otherwise, divide by denominator
' NOT ALL VALUES IN THE SENS 0 ROW WILL SEEM CORRECT
RunModelMenu - 3
'SOME OF THE HIGHER TIERS MAY ADD TO MORE THAN ONE
'BUT THE LOWEST MEASURES WILL HAVE THE CORRECT PROPORTION AND SUM TO ONE
For i = 40 To 2 Step -1
If (WeightSheet.Cells(SRow + 1, i)) > Tier Then
Power = WeightSheet.Cells(SRow + 1, i) - Tier
Child = 100 ^ Power
IsChild = (Int(WeightSheet.Cells(SRow + 2, i) / Child) - ID) = 0
If IsChild Then
WeightSheet.Cells(SRow + 8, i) = 0
Else
If Denom <> 0 Then ' Not to divide by zero
WeightSheet.Cells(SRow + 8, i) = WeightSheet.Cells(SRow + 7, i) / Denom
Else: WeightSheet.Cells(SRow + 8, i) = 0
End If
End If
Else ' Not in a lower Tier
If i <> Current And Denom <> 0 Then
WeightSheet.Cells(SRow + 8, i) = WeightSheet.Cells(SRow + 7, i) / Denom
Else
WeightSheet.Cells(SRow + 8, i) = 0
End If
End If
Next i
RefRow = 127 'Row for Global Measure Weights when Measure of interest = 0
Call ComputeSensitivityValues(RefRow, 0)
***** Compute Weights when Current Value is 1
'If a child in a lower tier, do not change
'If not a child, equal zero
'if on same tier then equal zero
'if higher tier, if a parent, weight is 1 otherwise 0
Denom = CurrentWt

```

```

For i = 40 To 2 Step -1
If (WeightSheet.Cells(SRow + 1, i)) > Tier Then
Power = WeightSheet.Cells(SRow + 1, i) - Tier
Child = 100 ^ Power
IsChild = (Int(WeightSheet.Cells(SRow + 2, i) / Child) - ID) = 0
If IsChild Then
If Denom <> 0 Then
WeightSheet.Cells(SRow + 9, i) = WeightSheet.Cells(SRow + 7, i) / Denom
Else: WeightSheet.Cells(SRow + 9, i) = 0
End If
Else
WeightSheet.Cells(SRow + 9, i) = 0
End If
Else ' Not in a lower Tier
If WeightSheet.Cells(SRow + 1, i) = Tier Then 'At Same Tier
If i = Current Then
WeightSheet.Cells(SRow + 9, i) = 1
Else
WeightSheet.Cells(SRow + 9, i) = 0
End If
Else: 'In a higher Tier
Power = Tier - WeightSheet.Cells(SRow + 1, i)
AParent = 100 ^ Power
IsAParent = (Int(ID / AParent) = WeightSheet.Cells(SRow + 2, i))
If IsAParent Then
WeightSheet.Cells(SRow + 9, i) = 1
Else: WeightSheet.Cells(SRow + 9, i) = 0
End If
End If ' If not in the Same Tier
End If 'If not in a lower Tier
Next i
RefRow = 128 'Row for Global Measure Weights when Measure of interest = 1
Call ComputeSensitivityValues(RefRow, 1)
End If ' If performing Local or Global Sensitivity Analysis
'Reset Weights Window
WeightSheet.Activate
ActiveWindow.ScrollRow = 1
RunModelMenu - 4
Range("B5").Select
'===== DRAW THE SENSITIVITY CHART
=====
countalt = Application.WorksheetFunction.CountA -
(Worksheets("Weights").Range("A132:A65356"))
sheetname = "Sensitivity"
Set SensSheet = Worksheets(sheetname)
Set ODSheet = Worksheets("Orig-Dest Airfields")
SensSheetActivate
SensSheet.Unprotect
OriginIndex = ODSheet.Cells(1, 2)
DestIndex = ODSheet.Cells(2, 2)
SensSheet.Range("L12") = ODSheet.Cells(FirstOD + OriginIndex - 1, 1)
SensSheet.Range("L15") = ODSheet.Cells(FirstOD + DestIndex - 1, 11)
SensSheet.Rows("5001:65356").Select
Selection.Delete Shift:=xlUp
ActiveSheet.ChartObjects.Delete 'Delete Previous Sheet
Set ch = SensSheet.ChartObjects.Add(5, 5, 475, 365)
ch.Name = "Chart1"
With SensSheet.DrawingObjects("Chart1")
.Placement = xlFreeFloating
.PrintObject = True
End With
SensSheet.ChartObjects("Chart1").Activate
SensSheet.ChartObjects("Chart1").Name = "Chart1"
ActiveChart.ChartType = xlXYScatterSmoothNoMarkers

```

```

ActiveChart.Location Where:=xlLocationAsObject, Name:=sheetname
ActiveChart.SeriesCollection.NewSeries
WtStr = Application.WorksheetFunction.Text(CurrentWt, "0.0000")
TitleStr = SensType & " Sensitivity Analysis on " & Analyze & vbCrLf & _
"Current weight is " & WtStr
With ActiveChart
    .HasTitle = True
    .Axes(xlCategory, xlPrimary).HasTitle = False
    .Axes(xlValue, xlPrimary).HasTitle = True
    .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Value"
    .ChartTitle.Characters.Text = TitleStr
End With
ActiveChart.HasLegend = True
'Fix value axis
With ActiveChart.Axes(xlValue)
    .MinimumScale = 0
    .MaximumScale = 1
    .MinorUnitIsAuto = True
    .MajorUnit = 0.1
    .Crosses = xlAutomatic
    .ReversePlotOrder = False
    .ScaleType = xlLinear
    .DisplayUnit = xlNone
End With
ActiveChart.Axes(xlValue).MajorGridlines.Select
With Selection.Border
    .ColorIndex = 57
    .Weight = xlHairline
    .LineStyle = xlDot
End With
With ActiveChart.Axes(xlCategory)
    .MinimumScale = 0
    .MaximumScale = 1
    .MinorUnitIsAuto = True
    .MajorUnit = 0.1
    .Crosses = xlAutomatic
    .ReversePlotOrder = False
    .ScaleType = xlLinear
    .DisplayUnit = xlNone
End With
RunModelMenu - 5
SensSheet.DrawingObjects("Chart1").RoundedCorners = True
SensSheet.DrawingObjects("Chart1").Shadow = True
'build data table by copying from Weights Sheet and graph
AltRange = "A132:C" & Application.WorksheetFunction.Text(131 + countalt, 0)
' Delete Previous data
WeightSheet.Range(AltRange).Copy _
ActiveSheet.Range("A5001")
'Build info for a Current Weight Line
SensSheet.Cells(5000, 4) = CurrentWt - 0.000001
SensSheet.Cells(5000, 5) = CurrentWt + 0.000001
SensSheet.Cells(5000 + countalt + 1, 1) = "Current Weight"
SensSheet.Cells(5000 + countalt + 1, 2) = ""
SensSheet.Cells(5000 + countalt + 1, 3) = ""
SensSheet.Cells(5000 + countalt + 1, 4) = -999
SensSheet.Cells(5000 + countalt + 1, 5) = 1000
DataRange = "A5000:E" & Application.WorksheetFunction.Text(5000 + countalt +
1, 0)
ActiveChart.SetSourceData Source:=Sheets(sheetname).Range(DataRange), _
PlotBy:=xlRows
' Make Lines Thick
For i = 1 To countalt
    ActiveChart.SeriesCollection(i).Select
    With Selection.Border

```

```

.Weight = xlThick
.LineStyle = xlContinuous
End With
Next i
ActiveChart.SeriesCollection(countalt + 1).Select
With Selection.Border
.ColorIndex = 3
End With
' Give Chart Area a Grey Background
ActiveChart.ChartArea.Select
Selection.Fill.OneColorGradient Style:=msoGradientHorizontal, Variant:=1, _
Degree:=0.231372549019608
With Selection
.Fill.Visible = True
.Fill.ForeColor.SchemeColor = 15
End With
ActiveChart.Legend.Select
Selection.Font.Size = 8
'Give Plot Area a Blue Background
ActiveChart.PlotArea.Select
Selection.Fill.TwoColorGradient Style:=msoGradientHorizontal, Variant:=2
With Selection
.Fill.Visible = True
.Fill.ForeColor.SchemeColor = 24
.Fill.BackColor.SchemeColor = 17
End With
' ===== END OF CHART DRAWING =====
WeightSheet.Protect
DataSheet.Protect
ValueSheet.Protect
SensSheet.Activate
SensSheet.Visible = True
ActiveSheet.PageSetup.PrintArea = "$A$1:$L$31"
RunModelMenu - 6
Worksheets("Main").Visible = False
Range("C4").Select
SensSheet.Protect
Application.ScreenUpdating = True
Unload Me
End Sub
Private Sub CommandButton3_Click()
Unload Me
End Sub
Private Sub CommandButton4_Click()
RunModelHelp.Show
End Sub
Private Sub CommandButton5_Click() 'Show ALL Scores
Application.ScreenUpdating = False
Call ComputeValueScores
Call ShowScores
Application.ScreenUpdating = True
Unload Me
End Sub
Private Sub OptionButton1_Click()
ByTier = 4
End Sub
Private Sub OptionButton2_Click()
ByTier = 3
End Sub
Private Sub OptionButton3_Click()
ByTier = 2
End Sub
Private Sub OptionButton4_Click()

```

```

ByTier = 1
End Sub
Private Sub OptionButton5_Click()
SensitivityOption = 1
End Sub
Private Sub OptionButton6_Click()
SensitivityOption = 2
End Sub
Private Sub UserForm_Initialize()
' Executed before the form is shown
Dim Col As Integer
Set DataSheet = ThisWorkbook.Sheets(DataSheetName)
Count = 39
For Col = 1 To Count
ComboBoxTopics.AddItem DataSheet.Cells(SRow, Col + 1)
Next Col
ComboBoxTopics.ListIndex = 0
Current = 2
Analyze = DataSheet.Cells(SRow, Current)
SensitivityOption = 1
ByTier = 1
Worksheets("Main").Activate
End Sub
Private Sub ComboBoxTopics_Click()
' Executed when the ComboBox is changed
Current = ComboBoxTopics.ListIndex + 2
Analyze = DataSheet.Cells(SRow, Current)
End Sub

SaveRestoreALL - 1
Dim OptionVal As Byte
Private Sub CommandButton1_Click()
Select Case OptionVal
Case 1
Msg = "This will overwrite the current default set" & vbCrLf _
& vbCrLf
& "Are you sure you want to Save As Default?"
Ans = MsgBox(Msg, vbYesNo, "Save As Default")
If (Ans = vbYes) Then
Call SaveALLDefaults
End If
Case 2
Msg = "Restoring defaults will reset to the default set" & vbCrLf _
& "value functions and weightings." & vbCrLf & vbCrLf -
& "Are you sure you want to Restore All Defaults?"
Ans = MsgBox(Msg, vbYesNo, "Restore Default")
If (Ans = vbYes) Then
Call RestoreALLDefaults
End If
Case 3
Msg = "Restoring the Original Model will reset ALL" & vbCrLf -
& "value functions and weightings." & vbCrLf & vbCrLf -
& "Are you sure you want to Restore Original Model?" -
Ans = MsgBox(Msg, vbYesNo, "Restore Original Model")
If (Ans = vbYes) Then
Call RestoreAllOriginals
End If
End Select
End Sub
Private Sub CommandButton2_Click()
Unload Me
End Sub
Private Sub OptionButton1_Click()
OptionVal = 1

```

```

End Sub
Private Sub OptionButton2_Click()
OptionVal = 2
End Sub
Private Sub OptionButton3_Click()
OptionVal = 3
End Sub
Private Sub UserForm_Click()
End Sub

SaveRestoreVF - 1
Dim OptionVal As Byte
Private Sub CommandButton1_Click()
Dim ContVF As Boolean
ContVF = GetContOrCat
If ContVF Then 'A Continuous Value Function
Select Case OptionVal
Case 1
Call SaveContVFDefault
Case 2
Call RestoreContVFDefault
Case 3
Call RestoreContVFOriginal
End Select
Else 'A Categorical VF
Select Case OptionVal
Case 1
Call SaveCatVFDefault
Case 2
Call RestoreCatVFDefault
Case 3
Call RestoreCatVFOriginal
End Select
End If
End Sub
Private Sub CommandButton2_Click()
Unload Me
End Sub
Private Sub OptionButton1_Click()
OptionVal = 1
End Sub
Private Sub OptionButton2_Click()
OptionVal = 2
End Sub
Private Sub OptionButton3_Click()
OptionVal = 3
End Sub
Private Sub UserForm_Click()
End Sub

SaveRestoreWeights - 1
Dim WOptVal As Byte
Private Sub CommandButton1_Click()
Select Case WOptVal
Case 1
Call SaveDefaultWeights
Case 2
Call RestoreDefaultWeights
Case 3
Call RestoreOriginalWeights
End Select
End Sub
Private Sub CommandButton2_Click()
Unload Me

```

```

End Sub
Private Sub OptionButton1_Click()
WOptVal = 1
End Sub
Private Sub OptionButton2_Click()
WOptVal = 2
End Sub
Private Sub OptionButton3_Click()
WOptVal = 3
End Sub
Private Sub UserForm_Click()
End Sub

ScenarioMenu - 1
Dim ValueWt, ProbabilityWt, NationalSecurityWt 'Current Weights for each
objective
Private Sub AirfieldBox_Change()
If AirfieldBox.ListIndex = -1 Then
MsgBox "You must select a valid airfield", vbOKOnly, "Invalid Input"
AirfieldBox.SetFocus
End If
End Sub
Private Sub ComboBox1_Change()
If ComboBox1.ListIndex = -1 Then
MsgBox "Must select a value from the drop down menu" & vbCrLf
& "This value must be changed for this form to be functional again", vbOKOnly,
"Invalid
d Input"
ComboBox1.SetFocus
End If
End Sub
Private Sub CommandButton10_Click()
HowToSwing.Show
End Sub
Private Sub CommandButton11_Click() 'Perform Sensitivity Analysis
'Weights are stored on the Weights Sheet, row 75
Set WeightSheet = Worksheets("Weights")
Dim Valid, SumtoOne As Boolean
WeightSheet.Unprotect
SumtoOne = WeightSheet.Cells(77, 2) 'Utilizes worksheet cell for summing the
current weights
'Determine that all weight text boxes have a valid input
Valid = ValueWt <> -1 And ProbabilityWt <> -1 And NationalSecurityWt <> -1 And
SumtoOne
If Not (Valid) Then
MsgBox "Please check to ensure all weights are numerical and sum to one." &
vbCrLf & "This may require 'swinging' the weights.", vbOKOnly, "Invalid Input"
Else
If AirfieldBox.ListIndex = -1 Then
MsgBox "You must select a valid airfield", vbOKOnly, "Invalid Input"
Else:
'Perform Sensitivity Analysis
OrigValueWt = WeightSheet.Cells(75, 2) 'B75
OrigProbabilityWt = WeightSheet.Cells(75, 3) 'C75
OrigNatSecurityWt = WeightSheet.Cells(75, 4) 'D75
CurrentWt = WeightSheet.Cells(75, 2 + ComboBox1.ListIndex)
Call InitializeSensitivity(CurrentWt) 'In the DestinationScenarios Module
'Run Scores for Weight of Interest = 0
'Maintain the original proportion between the remaining two objectives
Select Case ComboBox1.ListIndex
Case 0 'Enroute Value Score
Total = OrigProbabilityWt + OrigNatSecurityWt
WeightSheet.Cells(75, 2) = 0
WeightSheet.Cells(75, 3) = OrigProbabilityWt / Total

```

```

WeightSheet.Cells(75, 4) = OrigNatSecurityWt / Total
Case 1 'Probability
Total = OrigValueWt + OrigNatSecurityWt
WeightSheet.Cells(75, 2) = OrigValueWt / Total
WeightSheet.Cells(75, 3) = 0
WeightSheet.Cells(75, 4) = OrigNatSecurityWt / Total
Case 2 'National Security Index
Total = OrigValueWt + OrigProbabilityWt
WeightSheet.Cells(75, 2) = OrigValueWt / Total
WeightSheet.Cells(75, 3) = OrigProbabilityWt / Total
WeightSheet.Cells(75, 4) = 0
End Select
Call ComputeScenarioSensitivity(0)
WeightSheet.Unprotect
ScenarioMenu - 2
'Run Scores for Weight of Interest = 1
Select Case ComboBox1.ListIndex
Case 0 'Enroute Value Score
Total = OrigProbabilityWt + OrigNatSecurityWt
WeightSheet.Cells(75, 2) = 1
WeightSheet.Cells(75, 3) = 0
WeightSheet.Cells(75, 4) = 0
Case 1 'Probability
Total = OrigValueWt + OrigNatSecurityWt
WeightSheet.Cells(75, 2) = 0
WeightSheet.Cells(75, 3) = 1
WeightSheet.Cells(75, 4) = 0
Case 2 'National Security Index
Total = OrigValueWt + OrigProbabilityWt
WeightSheet.Cells(75, 2) = 0
WeightSheet.Cells(75, 3) = 0
WeightSheet.Cells(75, 4) = 1
End Select
Call ComputeScenarioSensitivity(1)
WeightSheet.Unprotect
'Restore Original Weights
WeightSheet.Cells(75, 2) = OrigValueWt
WeightSheet.Cells(75, 3) = OrigProbabilityWt
WeightSheet.Cells(75, 4) = OrigNatSecurityWt
Worksheets("Orig-Dest Airfields").Protect
Worksheets("Weights").Protect
Call DrawScenarioSensitivity(ComboBox1.Value)
Call ShowScenarioSensitivity
Unload Me
End If
End If
End Sub
Private Sub CommandButton2_Click()
Set WeightSheet = Worksheets("Weights")
Worksheets("Orig-Dest Airfields").Protect
WeightSheet.Protect
Unload Me
End Sub
Private Sub CommandButton1_Click() 'Show all Alternatives Scores
Set WeightSheet = Worksheets("Weights")
Dim Valid, SumtoOne As Boolean
SumtoOne = WeightSheet.Cells(77, 2) 'Uses worksheet function to find sum of
current weights
'Ensure that each weight input is valid and sum to one
Valid = ValueWt <> -1 And ProbabilityWt <> -1 And NationalSecurityWt <> -1 And
SumtoOne
If Not (Valid) Then
MsgBox "Please check to ensure all weights are numerical and sum to one." &
vbCrLf & "This may require 'swinging' the weights.", vbOKOnly, "Invalid Input"

```

```

Else
If AirfieldBox.ListIndex = -1 Then
MsgBox "You must select a valid airfield", vbOKOnly, "Invalid Input"
Else:
'Write valid weights to the Weights Sheet and sun the Scenario
Worksheets("Orig-Dest Airfields").Cells(1, 2) = AirfieldBox.ListIndex + 1
Worksheets("Orig-Dest Airfields").Protect
WeightSheet.Cells(75, 2) = ValueWt
WeightSheet.Cells(75, 3) = ProbabilityWt
WeightSheet.Cells(75, 4) = NationalSecurityWt
Worksheets("Orig-Dest Airfields").Protect
Worksheets("Weights").Protect
Call ShowScenarioMatrix
Unload Me
End If
End If
ScenarioMenu - 3
End Sub
Private Sub CommandButton3_Click() 'Show Top 10 Rankings
Set WeightSheet = Worksheets("Weights")
Dim Valid, SumtoOne As Boolean
SumtoOne = WeightSheet.Cells(77, 2)
'Ensure that each weight input is valid and sum to one
Valid = ValueWt <> -1 And ProbabilityWt <> -1 And NationalSecurityWt <> -1 And
SumtoOne
If Not (Valid) Then
MsgBox "Please check to ensure all weights are numerical and sum to one." &
vbCrLf & "This may require 'swinging' the weights.", vbOKOnly, "Invalid Input"
Else
If AirfieldBox.ListIndex = -1 Then
MsgBox "You must select a valid airfield", vbOKOnly, "Invalid Input"
Else:
'Write the valid weights to the weight sheet and call routine to create the
rankings chart
Worksheets("Orig-Dest Airfields").Cells(1, 2) = AirfieldBox.ListIndex + 1
Worksheets("Orig-Dest Airfields").Protect
WeightSheet.Cells(75, 2) = ValueWt
WeightSheet.Cells(75, 3) = ProbabilityWt
WeightSheet.Cells(75, 4) = NationalSecurityWt
Worksheets("Orig-Dest Airfields").Protect
Worksheets("Weights").Protect
Call ShowScenarioRankings
Unload Me
End If
End If
End Sub
Private Sub CommandButton8_Click()
HowToSwing.Show
End Sub
Private Sub CommandButton9_Click() 'Calculate Swing Weights
Set WeightSheet = Worksheets("Weights")
Dim Valid As Boolean
'Ensure all inputs are a valid number
Valid = ValueWt <> -1 And ProbabilityWt <> -1 And NationalSecurityWt <> -1
If Valid Then
Total = WeightSheet.Cells(76, 2)
ValueWt = WeightSheet.Cells(75, 2) / Total
ProbabilityWt = WeightSheet.Cells(75, 3) / Total
NationalSecurityWt = WeightSheet.Cells(75, 4) / Total
Wt1.Value = Round(ValueWt, 4)
Wt2.Value = Round(ProbabilityWt, 4)
Wt3.Value = Round(NationalSecurityWt, 4)
WeightSheet.Cells(75, 2) = ValueWt
WeightSheet.Cells(75, 3) = ProbabilityWt

```

```

WeightSheet.Cells(75, 4) = NationalSecurityWt
Else
MsgBox "Please check to ensure all weights are numerical.", vbOKOnly, "Invalid
Input"
End If
End Sub
Private Sub Frame1_Click()
End Sub
Private Sub Help_Click()
ScenarioMenuHelp.Show
End Sub
ScenarioMenu - 4
Private Sub Label10_Click()
Call ShowWF("Probability")
Unload Me
End Sub
Private Sub Label11_Click()
Call ShowWF("National Security")
Unload Me
End Sub
Private Sub Label8_Click()
MsgBox "There is no value function for the tactical value score", vbOKOnly,
"Value Score Value Fun
ction"
End Sub
Private Sub UserForm_Initialize()
Set ODSheet = Worksheets("Orig-Dest Airfields")
Set WeightSheet = Worksheets("Weights")
ODSheet.Unprotect
WeightSheet.Unprotect
ODSheetActivate
'Initialize Airfield Combo Box with list of Origins
Count = Application.WorksheetFunction.CountA(Range("A11:A65356"))
For Row = 1 To Count
AirfieldBox.AddItem ODSheet.Cells(Row + FirstOD - 1, 1)
Next Row
AirfieldBox.ListIndex = ODSheet.Cells(1, 2) - 1
'Initialize Weights text boxes from Weights Sheet row 75
Wt1.Value = Round(WeightSheet.Cells(75, 2), 3)
ValueWt = WeightSheet.Cells(75, 2)
Wt2.Value = Round(WeightSheet.Cells(75, 3), 3)
ProbabilityWt = WeightSheet.Cells(75, 3)
Wt3.Value = Round(WeightSheet.Cells(75, 4), 3)
NationalSecurityWt = WeightSheet.Cells(75, 4)
' Initialize entries for Sensitivity Analysis combo box
ComboBox1.AddItem "Tactical Sub-Model Score"
ComboBox1.AddItem "Probability of Utilization"
ComboBox1.AddItem "National Security Index"
ComboBox1.ListIndex = 0
Worksheets("Main").Activate
End Sub
Private Sub Wt1_Change()
Set WeightSheet = Worksheets("Weights")
ValueWt = ValidateWt(Wt1.Value)
If ValueWt <> -1 Then
WeightSheet.Cells(75, 2) = ValueWt
End If
End Sub
Private Sub Wt2_Change()
Set WeightSheet = Worksheets("Weights")
ProbabilityWt = ValidateWt(Wt2.Value)
If ProbabilityWt <> -1 Then
WeightSheet.Cells(75, 3) = ProbabilityWt
End If

```

```

End Sub
Private Sub Wt3_Change()
Set WeightSheet = Worksheets("Weights")
NationalSecurityWt = ValidateWt(Wt3.Value)
If NationalSecurityWt <> -1 Then
WeightSheet.Cells(75, 4) = NationalSecurityWt
ScenarioMenu - 5
End If
End Sub

ScenarioMenuHelp - 1
Private Sub CommandButton1_Click()
Unload Me
End Sub
Private Sub Label1_Click()
End Sub

SelectAirfield - 1
Dim AirfieldIndex As Long
Private Sub AirfieldBox_Change()
AirfieldIndex = AirfieldBox.ListIndex + 2
End Sub
Private Sub CommandButton1_Click()
Set USheet = Worksheets("Uncertainty Data")
Set DataSheet = Worksheets("Airfields Data")
If AirfieldIndex = 1 Then
MsgBox "You must select a valid airfield", vbOKOnly, "Invalid Input"
Else:
Select Case CommandButton1.Caption
Case "Select" ' Edit an Alternative
Unload Me
Call ShowEditAirfieldForm(AirfieldBox.Value, AirfieldIndex) 'In Alternatives
Module
Case "Remove"
Airfield = AirfieldBox.Value
Msg = "Are you sure you want to Remove " & Airfield & " ?"
Ans = MsgBox(Msg, vbYesNo, "Remove Airfield?")
If (Ans = vbYes) Then 'Delete airfield info in Airfields Data AND Uncertainty
Da
ta
DataSheet.Rows(AirfieldIndex).Delete
USheet.Unprotect
USheet.Rows(AirfieldIndex + 2).Delete
USheet.Protect
Unload Me
End If
End Select
End If
End Sub
Private Sub CommandButton2_Click()
Unload Me
End Sub
Private Sub UserForm_Initialize()
Count = Application.WorksheetFunction.CountA(Range("A2:A65356"))
For Row = 1 To Count
AirfieldBox.AddItem Cells(Row + 1, 1)
Next Row
AirfieldBox.ListIndex = 0
AirfieldIndex = 2
End Sub

SelectDestination - 1
Dim AirfieldIndex As Long
Dim Count As Integer

```

```

Private Sub AirfieldBox_Change()
AirfieldIndex = AirfieldBox.ListIndex
End Sub
Private Sub CommandButton1_Click()
If AirfieldIndex = -1 Then
MsgBox "You must select a valid airfield", vbOKOnly, "Invalid Input"
Else:
If CommandButton1.Caption = "Delete" Then
If Count = 1 Then
MsgBox "Cannot delete last origin. This will delete some formula
calculations." & vbCrLf
rLf &
"Suggest editing current airfield or adding new airfield", vbOKOnly, "Must hav
e one origin"
Unload Me
End If
Airfield = AirfieldBox.Value
Msg = "Are you sure you want to Remove " & Airfield & " ?"
Ans = MsgBox(Msg, vbYesNo, "Remove Airfield?")
If (Ans = vbYes) Then
ActiveSheet.Unprotect
AirfieldRow = Application.WorksheetFunction.Text(AirfieldIndex + FirstOD, 0)
SelectRange = "K" & AirfieldRow & ":AD" & AirfieldRow
Range(SelectRange).Select
Selection.Delete Shift:=xlUp
Range("A1").Select
ActiveSheet.Protect
Unload Me
End If
Else 'Else you are editing an Airfield
Row = AirfieldIndex + FirstOD
Load AddEditDestination
With AddEditDestination
.Caption = "Edit Destination Airfield"
.Airfield = Cells(Row, 11)
.Index = AirfieldIndex
.Probability.Value = Round(Cells(Row, 12) * 100, 2)
If Cells(Row, 13) = "?" Then
.CheckBox1 = True
.Label46.Visible = True
.Label47.Visible = True
.Label48.Visible = True
.NationalSecurityMode.Visible = True
.NationalSecurityMax.Visible = True
Else
.CheckBox1 = False
End If
.NationalSecurityMin.Value = Cells(Row, 14)
.NationalSecurityMode.Value = Cells(Row, 15)
.NationalSecurityMax.Value = Cells(Row, 16)
'.Throughput.Value = Cells(Row, 17)
.NorS.Value = Cells(Row, 18)
.LatDeg.Value = Cells(Row, 19)
.LatMin.Value = Cells(Row, 20)
.EorW.Value = Cells(Row, 21)
.LongDeg.Value = Cells(Row, 22)
.LongMin.Value = Cells(Row, 23)
'.CheckBox1 = True
.NationalSecurityMode.Enabled = True
.NationalSecurityMax.Enabled = True
End With
Unload Me
AddEditDestination.Show
End If

```

```

SelectDestination - 2
End If
End Sub
Private Sub CommandButton2_Click()
Unload Me
End Sub
Private Sub UserForm_Initialize()
Count = Application.WorksheetFunction.CountA(Range("K11:K65356"))
For Row = 1 To Count
AirfieldBox.AddItem Cells(Row + FirstOD - 1, 11)
Next Row
AirfieldBox.ListIndex = 0
AirfieldIndex = 0
End Sub

Sustainment - 1
Dim Weights(3), OldWeights(3)
Dim i As Byte
Private Sub CommandButton1_Click()
HowToSwing.Show
End Sub
Private Sub CommandButton2_Click()
Total = 0
' Swing the Weights
For i = 1 To 3
Total = Total + Weights(i)
Next i
For i = 1 To 3
Weights(i) = Weights(i) / Total
Next i
TextBox1.Value = Round(Weights(1), 4)
TextBox2.Value = Round(Weights(2), 4)
TextBox3.Value = Round(Weights(3), 4)
With Sheet3
.Range("G5") = Weights(1)
.Range("H5") = Weights(2)
.Range("I5") = Weights(3)
End With
Label7.Caption = Round(Sheet3.Range("G6"), 4)
Label8.Caption = Round(Sheet3.Range("H6"), 4)
Label9.Caption = Round(Sheet3.Range("I6"), 4)
End Sub
Private Sub CommandButton3_Click()
Dim Valid As Boolean
Valid = True
Total = 0
For i = 1 To 3
Valid = Valid And (Weights(i) <> -1)
Total = Total + Weights(i)
Next i
If (Not (Valid)) Or Not (EqualOne(Total)) Then
MsgBox "Please check to ensure all weights are numerical and sum to one." &
vbCrLf & "This may require 'swinging' the weights.", vbOKOnly, "Invalid Input"
Else:
With Sheet3
.Range("G5") = Weights(1)
.Range("H5") = Weights(2)
.Range("I5") = Weights(3)
End With
Unload Me
End If
End Sub
Private Sub CommandButton4_Click()
Sheet3.Range("G5") = OldWeights(1)

```

```

Sheet3.Range("H5") = OldWeights(2)
Sheet3.Range("I5") = OldWeights(3)
Unload Me
End Sub
Private Sub TextBox1_Change()
Weights(1) = ValidateWt(TextBox1.Value)
End Sub
Private Sub TextBox2_Change()
Weights(2) = ValidateWt(TextBox2.Value)
End Sub
Private Sub TextBox3_Change()
Weights(3) = ValidateWt(TextBox3.Value)
End Sub
Sustainment - 2
Private Sub UserForm_Initialize()
' Initialize Local Weights
OldWeights(1) = Sheet3.Range("G5")
OldWeights(2) = Sheet3.Range("H5")
OldWeights(3) = Sheet3.Range("I5")
TextBox1.Value = Round(OldWeights(1), 4)
TextBox2.Value = Round(OldWeights(2), 4)
TextBox3.Value = Round(OldWeights(3), 4)
' Initialize Global Weights
Label7.Caption = Round(Sheet3.Range("G6"), 4)
Label8.Caption = Round(Sheet3.Range("H6"), 4)
Label9.Caption = Round(Sheet3.Range("I6"), 4)
For i = 1 To 3
Weights(i) = OldWeights(i)
Next i
End Sub

TacSensitivityHelp - 1
Private Sub CommandButton1_Click()
Unload Me
End Sub

Throughput - 1
Dim Weights(3), OldWeights(3)
Dim i As Byte
Private Sub CommandButton1_Click()
HowToSwing.Show
End Sub
Private Sub CommandButton2_Click()
Total = 0
' Swing the Weights
For i = 1 To 3
Total = Total + Weights(i)
Next i
For i = 1 To 3
Weights(i) = Weights(i) / Total
Next i
TextBox1.Value = Round(Weights(1), 4)
TextBox2.Value = Round(Weights(2), 4)
TextBox3.Value = Round(Weights(3), 4)
With Sheet3
.Range("D5") = Weights(1)
.Range("E5") = Weights(2)
.Range("F5") = Weights(3)
End With
Label7.Caption = Round(Sheet3.Range("D6"), 4)
Label8.Caption = Round(Sheet3.Range("E6"), 4)
Label9.Caption = Round(Sheet3.Range("F6"), 4)
End Sub
Private Sub CommandButton3_Click()

```

```

Dim Valid As Boolean
Valid = True
Total = 0
For i = 1 To 3
    Valid = Valid And (Weights(i) <> -1)
    Total = Total + Weights(i)
Next i
If Not (Valid) Or Not (EqualOne(Total)) Then
    MsgBox "Please check to ensure all weights are numerical and sum to one." &
        vbCrLf & "This may require 'swinging' the weights.", vbOKOnly, "Invalid Input"
Else:
    With Sheet3
        .Range("D5") = Weights(1)
        .Range("E5") = Weights(2)
        .Range("F5") = Weights(3)
    End With
    Unload Me
End If
End Sub
Private Sub CommandButton4_Click()
    Sheet3.Range("D5") = OldWeights(1)
    Sheet3.Range("E5") = OldWeights(2)
    Sheet3.Range("F5") = OldWeights(3)
    Unload Me
End Sub
Private Sub TextBox1_Change()
    Weights(1) = ValidateWt(TextBox1.Value)
End Sub
Private Sub TextBox2_Change()
    Weights(2) = ValidateWt(TextBox2.Value)
End Sub
Private Sub TextBox3_Change()
    Weights(3) = ValidateWt(TextBox3.Value)
End Sub
Throughput = 2
Private Sub UserForm_Initialize()
    ' Initialize Local Weights
    OldWeights(1) = Sheet3.Range("D5")
    OldWeights(2) = Sheet3.Range("E5")
    OldWeights(3) = Sheet3.Range("F5")
    TextBox1.Value = Round(OldWeights(1), 4)
    TextBox2.Value = Round(OldWeights(2), 4)
    TextBox3.Value = Round(OldWeights(3), 4)
    ' Initialize Global Weights
    Label7.Caption = Round(Sheet3.Range("D6"), 4)
    Label8.Caption = Round(Sheet3.Range("E6"), 4)
    Label9.Caption = Round(Sheet3.Range("F6"), 4)
    For i = 1 To 3
        Weights(i) = OldWeights(i)
    Next i
End Sub

TriangleDist = 1
Private Sub CommandButton1_Click()
    Measure = Label2
    Set USheet = Worksheets("Uncertainty Data")
    Set WeightsSheet = Worksheets("Weights")
    'Test to ensure all inputs are numerical
    NotANum = False
    'Validate the Min
    Sheets("Weights").Range("B101") = Min.Value
    IsANumber = Sheets("Weights").Range("C101")
    If Not (IsANumber) Then
        NotANum = True

```

```

End If
'Validate the Mode
Sheets("Weights").Range("B101") = Mode.Value
IsANumber = Sheets("Weights").Range("C101")
If Not (IsANumber) Then
NotANum = True
End If
'Validate the Max
Sheets("Weights").Range("B101") = Max.Value
IsANumber = Sheets("Weights").Range("C101")
If Not (IsANumber) Then
NotANum = True
End If
If NotANum Then
MsgBox "One the values you have input is not a numerical value", vbOKOnly,
"Invalid Input"
NotANum = False
Else 'Go ahead and write the uncertainty values to the temp row
USheet.Unprotect
i = UCol
While USheet.Cells(1, i) <> Measure
i = i + 1
Wend
USheet.Cells(TempRow, i) = Min.Value
USheet.Cells(TempRow, i + 1) = Mode.Value
USheet.Cells(TempRow, i + 2) = Max.Value
USheet.Protect
Unload Me
End If
End Sub
Private Sub CommandButton2_Click()
Unload Me
End Sub
Private Sub UserForm_Click()
End Sub

Utilities - 1
Dim Weights(4), OldWeights(4)
Dim i As Byte
Private Sub CommandButton1_Click()
HowToSwing.Show
End Sub
Private Sub CommandButton2_Click()
Total = 0
'Swing the Weights
For i = 1 To 4
Total = Total + Weights(i)
Next i
For i = 1 To 4
Weights(i) = Weights(i) / Total
Next i
TextBox1.Value = Round(Weights(1), 4)
TextBox2.Value = Round(Weights(2), 4)
TextBox3.Value = Round(Weights(3), 4)
TextBox4.Value = Round(Weights(4), 4)
With Sheet3
.Range("E17") = Weights(1)
.Range("F17") = Weights(2)
.Range("G17") = Weights(3)
.Range("H17") = Weights(4)
End With
Label8.Caption = Round(Sheet3.Range("E18"), 4)
Label9.Caption = Round(Sheet3.Range("F18"), 4)
Label10.Caption = Round(Sheet3.Range("G18"), 4)

```

```

Label11.Caption = Round(Sheet3.Range("H18"), 4)
End Sub
Private Sub CommandButton3_Click()
Dim Valid As Boolean
Valid = True
Total = 0
For i = 1 To 4
Valid = Valid And (Weights(i) <> -1)
Total = Total + Weights(i)
Next i
If (Not (Valid)) Or Not (EqualOne(Total)) Then
MsgBox "Please check to ensure all weights are numerical and sum to one." &
vbCrLf & "This may require 'swinging' the weights.", vbOKOnly, "Invalid Input"
Else:
With Sheet3
.Range("E17") = Weights(1)
.Range("F17") = Weights(2)
.Range("G17") = Weights(3)
.Range("H17") = Weights(4)
End With
Unload Me
End If
End Sub
Private Sub CommandButton4_Click()
Sheet3.Range("E17") = OldWeights(1)
Sheet3.Range("F17") = OldWeights(2)
Sheet3.Range("G17") = OldWeights(3)
Sheet3.Range("H17") = OldWeights(4)
Unload Me
End Sub
Private Sub TextBox1_Change()
Weights(1) = ValidateWt(TextBox1.Value)
End Sub
Private Sub TextBox2_Change()
Weights(2) = ValidateWt(TextBox2.Value)
Utilities = 2
End Sub
Private Sub TextBox3_Change()
Weights(3) = ValidateWt(TextBox3.Value)
End Sub
Private Sub TextBox4_Change()
Weights(4) = ValidateWt(TextBox4.Value)
End Sub
Private Sub UserForm_Initialize()
' Initialize Local Weights
OldWeights(1) = Sheet3.Range("E17")
OldWeights(2) = Sheet3.Range("F17")
OldWeights(3) = Sheet3.Range("G17")
OldWeights(4) = Sheet3.Range("H17")
TextBox1.Value = Round(OldWeights(1), 4)
TextBox2.Value = Round(OldWeights(2), 4)
TextBox3.Value = Round(OldWeights(3), 4)
TextBox4.Value = Round(OldWeights(4), 4)
' Initialize Global Weights
Label8.Caption = Round(Sheet3.Range("E18"), 4)
Label9.Caption = Round(Sheet3.Range("F18"), 4)
Label10.Caption = Round(Sheet3.Range("G18"), 4)
Label11.Caption = Round(Sheet3.Range("H18"), 4)
For i = 1 To 4
Weights(i) = OldWeights(i)
Next i
End Sub

```

VFHelp - 1

```

Option Explicit
Dim TopicCount As Integer
Dim CurrentTopic As Integer
Dim HelpSheet As Worksheet
Const HelpSheetName As String = "VF Help"
Const HelpFormCaption As String = "Value Function Help"
Private Sub UserForm_Initialize()
    ' Executed before the form is shown
Dim Row As Integer
Set HelpSheet = ThisWorkbook.Sheets(HelpSheetName)
TopicCount = Application.WorksheetFunction.CountA(HelpSheet.Range("A:A"))
For Row = 1 To TopicCount
    ComboBoxTopics.AddItem HelpSheet.Cells(Row, 1)
Next Row
ComboBoxTopics.ListIndex = 0
If ActiveSheet.Range("L3") = "Score" Then
    CurrentTopic = 1
Else: CurrentTopic = 3
End If
UpdateForm
End Sub
Private Sub UpdateForm()
    ComboBoxTopics.ListIndex = CurrentTopic - 1
    Me.Caption = HelpFormCaption &
    "(" & CurrentTopic & " of " & TopicCount & ")"
    With LabelText
        .Caption = HelpSheet.Cells(CurrentTopic, 2)
        .AutoSize = False
        .Width = 220
        .AutoSize = True
    End With
    With Frame1
        .ScrollHeight = LabelText.Height + 5
        .ScrollTop = 1
    End With
    If CurrentTopic = 1 Then
        NextButton.Enabled = True
        NextButton.SetFocus
    ElseIf CurrentTopic = TopicCount Then
        PreviousButton.Enabled = True
        PreviousButton.SetFocus
    End If
    PreviousButton.Enabled = CurrentTopic <> 1
    NextButton.Enabled = CurrentTopic <> TopicCount
End Sub
Private Sub ComboBoxTopics_Click()
    ' Executed when the ComboBox is changed
    CurrentTopic = ComboBoxTopics.ListIndex + 1
    UpdateForm
End Sub
Private Sub PreviousButton_Click()
    ' Executed when the PreviousButton is clicked
    If CurrentTopic <> 1 Then
        CurrentTopic = CurrentTopic - 1
        UpdateForm
    End If
End Sub
Private Sub NextButton_Click()
    ' Executed when the NextButton is clicked
    If CurrentTopic <> TopicCount Then
        CurrentTopic = CurrentTopic + 1
        UpdateForm
    End If
End Sub

```

```
VFHelp - 2
Private Sub ExitButton_Click()
' Executed when the ExitButton is clicked
Unload Me
End Sub
```

Modules Code

```
Alternatives = 1
' AIRFIELD ALTERNATIVES MODULE
' This module contains subroutines for gathering Airfield Data
' Called by AirfieldInput UserForm
' Displays results on Airfields Data Worksheet
Public Const MCol As Byte = 7 'First Column where measures begin on Data Sheet
Sub ShowAlternativesHelp()
    AirfieldsHelp.Show
End Sub
Sub AddOrig()
    AddOrigin.Show
End Sub
Sub SelectOrigDest()
    DeleteOrig.Show
End Sub
Sub ShowAddDestination()
    ActiveSheet.Unprotect
    Load AddEditDestination 'Allows preloading of UserForm
    'The Caption is used in if-then statements in the AirfieldInput UserForm code
    AddEditDestination.Caption = "Add Destination Airfield"
    AddEditDestination.Show
    ActiveSheet.Protect
End Sub
Sub DeleteDestination()
    Load SelectDestination
    SelectDestination.Caption = "Delete Destination Airfield"
    SelectDestination.CommandButton1.Caption = "Delete"
    SelectDestination.Show
End Sub
Sub EditDestination()
    Load SelectDestination
    SelectDestination.Caption = "Edit Destination Airfield"
    SelectDestination.CommandButton1.Caption = "Edit"
    SelectDestination.Show
End Sub
Sub SortFields()
    ' Sorts alternatives on Airfields Data worksheet AND Uncertainty Data
    ' worksheet by the same Sort Keys
    ' If the decision is made to change the way the Airfields Data is sorted, then
    ' the Uncertainty Data must be sorted the same way
    ' This may require writing the same sort key to Uncertainty Data
    Set USheet = Worksheets("Uncertainty Data")
    Set DataSheet = Worksheets("Airfields Data")
    DataSheet.Activate
    DataSheet.Unprotect
    'This does count one alternative extra because of "Name" in Row 1
    Count = Application.WorksheetFunction.CountA(DataSheet.Range("A:A"))
    SortRange = "A2:AG" & Application.WorksheetFunction.Text(Count, 0)
    Range(SortRange).Select
    'Sort by Within the Lens "F1" then Name "A1"
    Selection.Sort Key1:=Range("F1"), Order1:=xlDescending, Key2:=Range("A1") _
    , Order2:=xlAscending, Header:=xlGuess, OrderCustom:=1, MatchCase:= _
    False, Orientation:=xlTopToBottom, DataOption1:=xlSortNormal, DataOption2 := _
    xlSortNormal
    Columns("A:AG").AutoFit
    Columns("I:I").ColumnWidth = 4.71
    Range("A1").Select
    DataSheet.Protect
    'Sort Uncertainty Data the same way
```

```

'Uncertainty Data Airfield index is +2 from Airfields Data due to column
headers
USheet.Unprotect
USheet.Activate
BottomRt = Application.WorksheetFunction.Text(Count + 2, 0)
SortRange = "A4:GN" & BottomRt
USheet.Range(SortRange).Select
Selection.Sort Key1:=Range("B3"), Order1:=xlDescending, Key2:=Range("A3") _
, Order2:=xlAscending, Header:=xlGuess, OrderCustom:=1, MatchCase:=
False, Orientation:=xlTopToBottom, DataOption1:=xlSortNormal, DataOption2 :=
:=xlSortNormal
Alternatives - 2
USheet.Columns("A:GN").AutoFit
FormatRange = "O3:BB" & BottomRt & ",B03:GN" & BottomRt
USheet.Range(FormatRange).Select
Selection.NumberFormat = "0.000"
USheet.Range("A1").Select
USheet.Protect
DataSheetActivate
End Sub
Sub SortOrigDest()
Worksheets("Orig-Dest Airfields").Activate
ActiveSheet.Unprotect
'Sort Origins
Count = Application.WorksheetFunction.CountA(Worksheets("Orig-Dest
Airfields").Range("A1:A65356"))
)
SortRange = "A11:I" & Application.WorksheetFunction.Text(Count + 11, 0)
Range(SortRange).Select
'Sort OrigDest by E/W then by Longitude
Selection.Sort Key1:=Range("E10"), Order1:=xlDescending, Key2:=Range( _
"F10"), Order2:=xlAscending, Header:=xlGuess, OrderCustom:=1, MatchCase :=
:=False, Orientation:=xlTopToBottom, DataOption1:=xlSortNormal, _
DataOption2:=xlSortNormal
Columns("A:I").AutoFit
'Sort Destinations
Count = Application.WorksheetFunction.CountA(Worksheets("Orig-Dest
Airfields").Range("K11:K65356"))
)
SortRange = "K11:AD" & Application.WorksheetFunction.Text(Count + 11, 0)
Range(SortRange).Select
'Sort OrigDest by E/W then by Longitude
Selection.Sort Key1:=Range("U10"), Order1:=xlDescending, Key2:=Range( _
"V10"), Order2:=xlAscending, Header:=xlGuess, OrderCustom:=1, MatchCase :=
:=False, Orientation:=xlTopToBottom, DataOption1:=xlSortNormal, _
DataOption2:=xlSortNormal
Columns("K:AD").AutoFit
Columns("L:AD").HorizontalAlignment = xlCenter
Range("A1").Select
ActiveSheet.Protect
End Sub
Sub ShowAddAirfieldForm()
Application.ScreenUpdating = False
ActiveSheet.Unprotect
Load AirfieldInput 'Allows preloading of UserForm
'The Caption is used in if-then statements in the AirfieldInput UserForm code
AirfieldInput.Caption = "Add a new airfield alternative"
Call InitializeTempRow ' Prepares initial uncertainty data for new
alternative, in Uncert
ainty Module
Application.ScreenUpdating = True
AirfieldInput.Show
ActiveSheet.Protect
End Sub

```

```

Sub SelectEditAirfield()
Application.ScreenUpdating = False
ActiveSheet.Unprotect
Load SelectAirfield 'Same Userform used for Removing airfield
SelectAirfield.Caption = "Select Airfield to Edit" 'To decipher from an edit
or remove
SelectAirfield.CommandButton1.Caption = "Select"
Application.ScreenUpdating = True
SelectAirfield.Show
ActiveSheet.Protect
End Sub
Sub RemoveAirfield()
ActiveSheet.Unprotect
Load SelectAirfield 'Same userform used for Editing airfield
SelectAirfield.Caption = "Select Airfield to Remove" 'Deciphers between edit
or remove
SelectAirfield.CommandButton1.Caption = "Remove"
SelectAirfield.Show
Alternatives - 3
ActiveSheet.Protect
End Sub
Sub ShowEditAirfieldForm(AirfieldName, AirfieldIndex)
Application.ScreenUpdating = False
'Called by SelectAirfield UserForm
i = AirfieldIndex
Worksheets("Airfields Data").Activate
ActiveSheet.Unprotect
Load AirfieldInput ' LOTS OF CODE in AirfieldInput UserForm
Set AI = AirfieldInput
' With AirfieldInput
AI.Caption = "Edit airfield alternative"
AI.Airfield.Value = AirfieldName
AI.Index = i
AI.NorS.Value = Cells(i, 2)
'Convert back from Degree-Decimal to Degree-Minute-Decimal format
AI.LatDeg.Value = Abs(Application.WorksheetFunction.Floor(Abs(Cells(i, 3)),
1))
AI.LatMin.Value = Round(Abs((Cells(i, 3) - AirfieldInput.LatDeg.Value) * 60),
2)
AI.EorW.Value = Cells(i, 4)
AI.LongDeg.Value = Abs(Application.WorksheetFunction.Floor(Abs(Cells(i, 5)),
1))
AI.LongMin.Value = Round(Abs((Cells(i, 5) - AirfieldInput.LongDeg.Value) *
60), 2)
'=====
' Load the data for each measure
' If the an uncertain probability is assigned to a measure, signified by the
"?" in the data field
' Then the measure is loaded with the checkbox marked and the input box
disabled
' If no uncertainty is assigned, the value is loaded
' Repeat for all 27 measures (except Critical Leg and Delta Flight Length)
If Cells(i, MCol) = "?" Then
AI.MOG.Enabled = False
AI.MOG.BackColor = &H8000000F
AI.CheckBox1 = True
Else
AI.MOG.Value = Cells(i, MCol)
End If
If Cells(i, MCol + 3) = "?" Then
AI.AlternateAirfields.Enabled = False
AI.AlternateAirfields.BackColor = &H8000000F
AI.CheckBox2 = True

```

```

Else
AI.AlternateAirfields.Value = Cells(i, MCol + 3)
End If
If Cells(i, MCol + 4) = "?" Then
AI.FuelStorage.Enabled = False
AI.FuelStorage.BackColor = &H8000000F
AI.CheckBox3 = True
Else
AI.FuelStorage.Value = Cells(i, MCol + 4)
End If
If Cells(i, MCol + 5) = "?" Then
AI.FuelResupply.Enabled = False
AI.FuelResupply.BackColor = &H8000000F
AI.CheckBox4 = True
Else
AI.FuelResupply.Value = Cells(i, MCol + 5)
End If
If Cells(i, MCol + 6) = "?" Then
AI.DiplomaticClearance.Enabled = False
AI.DiplomaticClearance.BackColor = &H8000000F
AI.CheckBox5 = True
Else
AI.DiplomaticClearance.Value = Cells(i, MCol + 6)
Alternatives - 4
End If
If Cells(i, MCol + 7) = "?" Then
AI.ForceProtection.Enabled = False
AI.ForceProtection.BackColor = &H8000000F
AI.CheckBox6 = True
Else
AI.ForceProtection.Value = Cells(i, MCol + 7)
End If
If Cells(i, MCol + 8) = "?" Then
AI.DeptOfState.Enabled = False
AI.DeptOfState.BackColor = &H8000000F
AI.CheckBox7 = True
Else
AI.DeptOfState.Value = Cells(i, MCol + 8)
End If
If Cells(i, MCol + 9) = "?" Then
AI.MilitaryCooperation.Enabled = False
AI.MilitaryCooperation.BackColor = &H8000000F
AI.CheckBox8 = True
Else
AI.MilitaryCooperation.Value = Cells(i, MCol + 9)
End If
If Cells(i, MCol + 10) = "?" Then
AI.Seaport.Enabled = False
AI.Seaport.BackColor = &H8000000F
AI.CheckBox9 = True
Else
AI.Seaport.Value = Cells(i, MCol + 10)
End If
If Cells(i, MCol + 11) = "?" Then
AI.Railroad.Enabled = False
AI.Railroad.BackColor = &H8000000F
AI.CheckBox10 = True
Else
AI.Railroad.Value = Cells(i, MCol + 11)
End If
If Cells(i, MCol + 12) = "?" Then
AI.RoadSystems.Enabled = False
AI.RoadSystems.BackColor = &H8000000F
AI.CheckBox11 = True

```

```

Else
AI.RoadSystems.Value = Cells(i, MCol + 12)
End If
If Cells(i, MCol + 13) = "?" Then
AI.CommercialAirport.Enabled = False
AI.CommercialAirport.BackColor = &H8000000F
AI.CheckBox12 = True
Else
AI.CommercialAirport.Value = Cells(i, MCol + 13)
End If
If Cells(i, MCol + 14) = "?" Then
AI.Lodging.Enabled = False
AI.Lodging.BackColor = &H8000000F
AI.CheckBox13 = True
Else
AI.Lodging.Value = Cells(i, MCol + 14)
End If
If Cells(i, MCol + 15) = "?" Then
AI.Dining.Enabled = False
AI.Dining.BackColor = &H8000000F
AI.CheckBox14 = True
Else
AI.Dining.Value = Cells(i, MCol + 15)
End If
Alternatives - 5
If Cells(i, MCol + 16) = "?" Then
AI.Medical.Enabled = False
AI.Medical.BackColor = &H8000000F
AI.CheckBox15 = True
Else
AI.Medical.Value = Cells(i, MCol + 16)
End If
If Cells(i, MCol + 17) = "?" Then
AI.Comm.Enabled = False
AI.Comm.BackColor = &H8000000F
AI.CheckBox16 = True
Else
AI.Comm.Value = Cells(i, MCol + 17)
End If
If Cells(i, MCol + 18) = "?" Then
AI.Power.Enabled = False
AI.Power.BackColor = &H8000000F
AI.CheckBox17 = True
Else
AI.Power.Value = Cells(i, MCol + 18)
End If
If Cells(i, MCol + 19) = "?" Then
AI.Water.Enabled = False
AI.Water.BackColor = &H8000000F
AI.CheckBox18 = True
Else
AI.Water.Value = Cells(i, MCol + 19)
End If
If Cells(i, MCol + 20) = "?" Then
AI.Sewer.Enabled = False
AI.Sewer.BackColor = &H8000000F
AI.CheckBox19 = True
Else
AI.Sewer.Value = Cells(i, MCol + 20)
End If
If Cells(i, MCol + 21) = "?" Then
AI.Mountainous.Enabled = False
AI.Mountainous.BackColor = &H8000000F
AI.CheckBox20 = True

```

```

Else
AI.Mountainous.Value = Cells(i, MCol + 21)
End If
If Cells(i, MCol + 22) = "?" Then
AI.Altitude.Enabled = False
AI.Altitude.BackColor = &H8000000F
AI.CheckBox21 = True
Else
AI.Altitude.Value = Cells(i, MCol + 22)
End If
If Cells(i, MCol + 23) = "?" Then
AI.Weather.Enabled = False
AI.Weather.BackColor = &H8000000F
AI.CheckBox22 = True
Else
AI.Weather.Value = Cells(i, MCol + 23)
End If
If Cells(i, MCol + 24) = "?" Then
AI.Temperature.Enabled = False
AI.Temperature.BackColor = &H8000000F
AI.CheckBox23 = True
Else
AI.Temperature.Value = Cells(i, MCol + 24)
End If
Alternatives - 6
If Cells(i, MCol + 25) = "?" Then
AI.UrbanAreas.Enabled = False
AI.UrbanAreas.BackColor = &H8000000F
AI.CheckBox24 = True
Else
AI.UrbanAreas.Value = Cells(i, MCol + 25)
End If
If Cells(i, MCol + 26) = "?" Then
AI.Terrain.Enabled = False
AI.Terrain.BackColor = &H8000000F
AI.CheckBox25 = True
Else
AI.Terrain.Value = Cells(i, MCol + 26)
End If
=====
=====

'Copy uncertainty data to TempRow
Set USheet = Worksheets("Uncertainty Data")
USheet.Unprotect
USheet.Rows(i + 2).Copy USheet.Rows(TempRow)
USheet.Protect
Application.ScreenUpdating = True
AirfieldInput.Show
End Sub

Function ValidContData(Val) As Boolean
' Called by AirfieldInput UserForm to verify numerical input to Continuous
measures
ValidContData = False
'Uses unlocked cells in Weights sheet to utilize the worksheet function
IsNumber
Sheets("Weights").Range("B101") = Val
IsWtANumber = Sheets("Weights").Range("C101")
ValIsZero = (Val = "") Or (Val = ".")
If IsWtANumber Or ValIsZero Then
ValidContData = True
End If
End Function

Sub GetContVFRRange(Measure, lo, hi)
'Called by AirfieldInput UserForm to show ControlTipText

```

```

' Also used in Distance Calculations to determine max range for Critical Leg
lens
' Even if function is decreasing, lo will still be less than hi
Worksheets(Measure).Unprotect
lo = Worksheets(Measure).Range("A100")
hi = Worksheets(Measure).Range("A110")
Worksheets(Measure).Protect
If lo > hi Then
Temp = lo
lo = hi
hi = Temp
End If
End Sub

Defaults - 1
' DEFAULTS MODULE
' Modules for Setting and Restoring Original DEFAULT values for functions and
weights
Public Const MeasureSheets = 21 ' First Sheet where Measure Sheets begin
Function GetContOrCat() As Boolean
' Used only in this module and by SaveRestoreVF UserForm
' Returns TRUE for a Continuous VF Sheet, FALSE for a Categorical VF Sheet
GetContOrCat = (Range("L3") = "Score")
End Function
Sub RedrawAfterContVFUpdate()
'After Copying Default/Original upper/lower bounds back to the current sheet,
the borders need to be r
edrawn
Application.ScreenUpdating = False
Range("M14:M15").Select
With Selection
.HorizontalAlignment = xlCenter
.Font.Size = 12
End With
With Selection.Borders(xlEdgeLeft)
.LineStyle = xlContinuous
.Weight = xlThin
.ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeTop)
.LineStyle = xlContinuous
.Weight = xlThin
.ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeBottom)
.LineStyle = xlContinuous
.Weight = xlThin
.ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeRight)
.LineStyle = xlContinuous
.Weight = xlThin
.ColorIndex = xlAutomatic
End With
With Selection.Borders(xlInsideHorizontal)
.LineStyle = xlContinuous
.Weight = xlThin
.ColorIndex = xlAutomatic
End With
Range("L14:M15").Select
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeLeft)
.LineStyle = xlContinuous

```

```

.Weight = xlMedium
.ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeTop)
.LineStyle = xlContinuous
.Weight = xlMedium
.ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeBottom)
.LineStyle = xlContinuous
.Weight = xlMedium
.ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeRight)
.LineStyle = xlContinuous
.Weight = xlMedium
.ColorIndex = xlAutomatic
End With
With Selection.Borders(xlInsideVertical)
.LineStyle = xlContinuous
.Weight = xlThin
.ColorIndex = xlAutomatic
Defaults - 2
End With
With Selection.Borders(xlInsideHorizontal)
.LineStyle = xlContinuous
.Weight = xlThin
.ColorIndex = xlAutomatic
End With
Call Adjust_Graph_new_bounds 'Ensures that the Graph bounds matches the latest
inputs
Application.ScreenUpdating = True
End Sub
Sub SaveContVFDefault()
' Copies the Current Chart information to rows 89:91 as the Default info
ActiveSheet.Unprotect
For j = 1 To 15 Step 2
ActiveSheet.Cells(89, j) = ActiveSheet.Cells(99, j)
ActiveSheet.Cells(91, j) = ActiveSheet.Cells(110, j)
Next j
Range("A90") = Range("A100")
ActiveSheet.Protect DrawingObjects:=True, Contents:=True, Scenarios:=True
ActiveSheet.EnableSelection = xlUnlockedCells
End Sub
Sub RestoreContVFDefault()
'
' RestoreContVFDefault Macro
' Macro recorded 2/18/2006 by Alex Miravite
'Copies the Default info (rows 89-91) to the current chart values that start
at row 99
'Redraws the upper/lower bounds adjust the chart again
ActiveSheet.Unprotect
For j = 1 To 15 Step 2
ActiveSheet.Cells(99, j) = ActiveSheet.Cells(89, j)
ActiveSheet.Cells(110, j) = ActiveSheet.Cells(91, j)
Next j
Range("A100") = Range("A90")
Range("A100").Copy Range("M14")
Range("A110").Copy Range("M15")
Range("B115:B135").Copy Range("A115")
Call RedrawAfterContVFUpdate
ActiveSheet.Protect DrawingObjects:=True, Contents:=True, Scenarios:=True
ActiveSheet.EnableSelection = xlUnlockedCells
End Sub

```

```

Sub RestoreContVFOriginal()
' Copies Original VF information from rows 79-81 to the current chart values
starting at row 99
' Updates the upper/lower bounds and redraws chart
ActiveSheet.Unprotect
For j = 1 To 15 Step 2
ActiveSheet.Cells(99, j) = ActiveSheet.Cells(79, j)
ActiveSheet.Cells(110, j) = ActiveSheet.Cells(81, j)
Next j
Range("A100") = Range("A80")
Range("A100").Copy Range("M14")
Range("A110").Copy Range("M15")
Range("C115:C135").Copy Range("A115")
Call RedrawAfterContVFUpdate
ActiveSheet.Protect DrawingObjects:=True, Contents:=True, Scenarios:=True
ActiveSheet.EnableSelection = xlUnlockedCells
Defaults - 3
End Sub
Sub SaveRestoreVFMenu() 'Displayed when the command button is clicked
SaveRestoreVF.Show
End Sub
Sub SaveCatVFDefault()
' Copies the current categorical functon values to row 100 as default
ActiveSheet.Unprotect
Range("L3:M12").Copy Range("L100")
ActiveSheet.Protect DrawingObjects:=True, Contents:=True, Scenarios:=True
ActiveSheet.EnableSelection = xlUnlockedCells
End Sub
Sub RestoreCatVFDefault()
'
' RestoreCategoricalVFDefaults Macro
' Macro recorded 2/18/2006 by Alex Miravite
'
' Copies the Default values from Row 100 to the current categorical VF values
ActiveSheet.Unprotect
Range("L100:M109").Copy Range("L3")
Call Adjust_graph_category_add_or_delete
ActiveSheet.Protect DrawingObjects:=True, Contents:=True, Scenarios:=True
ActiveSheet.EnableSelection = xlUnlockedCells
End Sub
Sub RestoreCatVFOriginal()
'Copies the Original Categorical VF values stored at row 80 back to the
current values
ActiveSheet.Unprotect
Range("L80:M89").Copy Range("L3")
Call Adjust_graph_category_add_or_delete
ActiveSheet.Protect DrawingObjects:=True, Contents:=True, Scenarios:=True
ActiveSheet.EnableSelection = xlUnlockedCells
End Sub
Sub SaveRestoreWeightsMenu()
' Attached to the command button on the Weights sheet
SaveRestoreWeights.Show
End Sub
Function ValidateWt(Val)
' Used in the swing weighting forms
' Returns the value if the weight is valid, -1 if it is invalid
' Utilized worksheet function IsNumber in Worksheet "Weights" cells B101 and
C101
Dim IsWtANumber As Boolean
Sheets("Weights").Range("B101") = Val
IsWtANumber = Sheets("Weights").Range("C101")
ValIsZero = (Val = "") Or Val = "."
If IsWtANumber Or ValIsZero Then
If ValIsZero Then

```

```

ValidateWt = 0
' MsgBox "Val " & Val & " Val Zero? " & ValIsZero & " ValidateWt " & ValidateWt
Else
If Val >= 0 Then
ValidateWt = Val
Else:
MsgBox "Weights must be input as a positive number." & vbCrLf & "Previous
value will b
e used", vbOKOnly, "Invalid Input"
ValidateWt = -1
End If
End If
Else:
MsgBox "Weights must be input as a postive numer." & vbCrLf & "Previous value
will be used", v
bOKOnly, "Invalid Input"
ValidateWt = -1
End If
Defaults = 4
End Function
Sub SaveDefaultWeights()
'
' SetDefaultWeights Macro
' Macro recorded 2/18/2006 by Alex Miravite
'

' Saves the LOCAL weights to the default data location starting in row 94
' All GLOBAL weights are calculated by the formulas in the visible section of
the Weights sheet
Application.ScreenUpdating = False
Worksheets("Weights").Activate
ActiveSheet.Unprotect
Range("B5:I5").Copy Range("B94")
Range("B9:J9").Copy Range("B95")
Range("B13:J13").Copy Range("B96")
Range("B17:H17").Copy Range("B97")
Range("B21:G21").Copy Range("B98")
Range("B94:J98").Select
Selection.Borders(xlEdgeLeft).LineStyle = xlNone
Selection.Borders(xlEdgeTop).LineStyle = xlNone
Selection.Borders(xlEdgeBottom).LineStyle = xlNone
Selection.Borders(xlEdgeRight).LineStyle = xlNone
Selection.Borders(xlInsideVertical).LineStyle = xlNone
Selection.Borders(xlInsideHorizontal).LineStyle = xlNone
ActiveWindow.ScrollRow = 1
Range("B5").Select
ActiveSheet.Protect DrawingObjects:=True, Contents:=True, Scenarios:=True
ActiveSheet.EnableSelection = xlUnlockedCells
Application.ScreenUpdating = True
End Sub
Sub RestoreDefaultWeights()
'
' RestoreDefaultWeights Macro
' Macro recorded 2/18/2006 by Alex Miravite
'

' Restores the values saved as the default weights starting in Row 94
'

Application.ScreenUpdating = False
Worksheets("Weights").Activate
ActiveSheet.Unprotect
Range("B94:I94").Copy
Range("B5").Select
Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _ :=False, Transpose:=False
Range("B95:J95").Copy

```

```

Range("B9").Select
Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _ 
:=False, Transpose:=False
Range("B96:J96").Copy
Range("B13").Select
Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _ 
:=False, Transpose:=False
Range("B97:H97").Copy
Range("B17").Select
Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _ 
:=False, Transpose:=False
Range("B98:G98").Copy
Range("B21").Select
Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _ 
:=False, Transpose:=False
ActiveWindow.ScrollRow = 1
Range("B5").Select
ActiveSheet.Protect DrawingObjects:=True, Contents:=True, Scenarios:=True
ActiveSheet.EnableSelection = xlUnlockedCells
Defaults - 5
Application.ScreenUpdating = True
End Sub
Sub RestoreOriginalWeights()
'
' Restores Weights originally determined by AFIT/ENS
'
' Restores the originally set weights stored in row 84 and below
Application.ScreenUpdating = False
Worksheets("Weights").Activate
ActiveSheet.Unprotect
Range("B84:I84").Copy
Range("B5").Select
Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _ 
:=False, Transpose:=False
Range("B85:J85").Copy
Range("B9").Select
Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _ 
:=False, Transpose:=False
Range("B86:J86").Copy
Range("B13").Select
Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _ 
:=False, Transpose:=False
Range("B87:H87").Copy
Range("B17").Select
Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _ 
:=False, Transpose:=False
Range("B88:G88").Copy
Range("B21").Select
Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _ 
:=False, Transpose:=False
ActiveWindow.ScrollRow = 1
Range("B5").Select
ActiveSheet.Protect DrawingObjects:=True, Contents:=True, Scenarios:=True
ActiveSheet.EnableSelection = xlUnlockedCells
Application.ScreenUpdating = True
End Sub
Sub SaveRestoreALLMenu()
' Attached to the textbox on the Main Menu
SaveRestoreALL.Show
End Sub
Sub SaveALLDefaults()
Dim ContVF As Boolean
Call SaveDefaultWeights
Application.ScreenUpdating = False

```

```

For i = MeasureSheets To Sheets.Count
Sheets(i).Activate
ContVF = GetContOrCat
If ContVF Then ' A Continuous VF sheet
Call SaveContVFDefault
Else: Call SaveCatVFDefault ' A Categorical VF Sheet
End If
Next i
Sheets(1).Activate
Application.ScreenUpdating = True
End Sub
Sub RestoreALLDefaults()
Dim ContVF As Boolean
Defaults - 6
Call RestoreDefaultWeights
Application.ScreenUpdating = False
For i = MeasureSheets To Sheets.Count
Sheets(i).Activate
ContVF = GetContOrCat
If ContVF Then ' A Continuous VF sheet
Call RestoreContVFDefault
Else: Call RestoreCatVFDefault ' A Categorical VF Sheet
End If
Next i
Sheets(1).Activate
Application.ScreenUpdating = True
End Sub
Sub RestoreALLOriginals()
Dim ContVF As Boolean
Call RestoreOriginalWeights
Application.ScreenUpdating = False
For i = MeasureSheets To Sheets.Count
Sheets(i).Activate
ContVF = GetContOrCat
If ContVF Then ' A Continuous VF sheet
Call RestoreContVFOriginal
Else: Call RestoreCatVFOriginal ' A Categorical VF Sheet
End If
Next i
Sheets(1).Activate
Application.ScreenUpdating = True
End Sub

DestinationScenarios - 1
'DESTINATION SCENARIO MODULE
' This module allows an origin to be selected that will be paired with all
possible destinations
' Each alternative that is within the lens will be scored and factored by the
Probability of travel to
the destination
' and the level of National Security interest
' The resulting scores for each alternative will be sorted and charted
Public Const VMCol As Byte = 4 ' First Column of measures on Value Sheet
Public Const DMCOL As Byte = 7 ' First Column of measures on Data Sheet
Public ScoresDisplayed As Boolean ' keeps track of whether scores are
displayed on chart or not
Sub ComputeScenario()
Set ValueSheet = Worksheets("Values")
Set DataSheet = Worksheets("Airfields Data")
Set ScoreSheet = Worksheets("Scenario Matrix")
Set WeightSheet = Worksheets("Weights")
Set ODSheet = Worksheets("Orig-Dest Airfields")
ValueSheet.Unprotect
DataSheet.Unprotect

```

```

ScoreSheet.Unprotect
WeightSheet.Unprotect
ScoreSheet.Activate
Application.ScreenUpdating = False
'Initialize Scenario Matrix
Call InitializeMatrix
'Get Origin Index
OriginIndex = ODSheet.Cells(1, 2)
OldDestIndex = ODSheet.Cells(2, 2)
NumDest = Application.WorksheetFunction.CountA(ODSheet.Range("K11:K65356"))
j = 4 'Uses the same row in the Values sheet to accomplish calculations
'Delete all current values in the value sheet
ValueSheet.Rows("5:65356").Delete
For dest = 1 To NumDest
    ODSheet.Cells(2, 2) = dest
    Call ComputeDistances
        ' This subroutine will compute the distances for the current OD pair
        ' All alternatives will be sorted by with those in the lens at the top
        'Next, determine if current OD pair requires an Enroute
    DistOtoD = ODSheet.Cells(7, 9) ' Cell I7
    If EnrouteRequired(DistOtoD) Then 'An enroute is required, so compute scenario
    scores
    for valid alternatives
        Alternative = 1
        ' While DataSheet.Cells(Alternative + 1, 6) <> False 'OPTION: While looking at
        alternatives in lens
        While DataSheet.Cells(Alternative + 1, 1) <> "" ' Look at all alternatives
            If DataSheet.Cells(Alternative + 1, 6) <> False Then 'If in the Lens
                'Compute the Value Scores
                i = Alternative 'Row Index for Data Sheet for current alternative
                j = j + 1 'Row Index where to write result in Value Sheet
                ValueSheet.Cells(j, 2) = DataSheet.Cells(i + 1, 1)
                Call ComputeMeasureValue(i, j, "MOG", 0)
                Call ComputeMeasureValue(i, j, "Critical Leg", 1)
                Call ComputeMeasureValue(i, j, "Delta Flight Length", 2)
                Call ComputeMeasureValue(i, j, "Alternate Airfields", 3)
                Call ComputeMeasureValue(i, j, "Fuel Storage", 4)
                Call ComputeMeasureValue(i, j, "Fuel Resupply", 5)
                DestinationScenarios - 2
                Call ComputeMeasureValue(i, j, "Diplomatic Clearance", 6)
                Call ComputeMeasureValue(i, j, "Force Protection", 7)
                Call ComputeMeasureValue(i, j, "Dept of State", 8)
                Call ComputeMeasureValue(i, j, "Military Cooperation", 9)
                Call ComputeMeasureValue(i, j, "Seaport", 10)
                Call ComputeMeasureValue(i, j, "Railroad", 11)
                Call ComputeMeasureValue(i, j, "Road System", 12)
                Call ComputeMeasureValue(i, j, "Commercial Airport", 13)
                Call ComputeMeasureValue(i, j, "Lodging", 14)
                Call ComputeMeasureValue(i, j, "Dining", 15)
                Call ComputeMeasureValue(i, j, "Medical", 16)
                Call ComputeMeasureValue(i, j, "Communications", 17)
                Call ComputeMeasureValue(i, j, "Power", 18)
                Call ComputeMeasureValue(i, j, "Potable Water", 19)
                Call ComputeMeasureValue(i, j, "Sewer", 20)
                Call ComputeMeasureValue(i, j, "Mountainous", 21)
                Call ComputeMeasureValue(i, j, "Altitude", 22)
                Call ComputeMeasureValue(i, j, "Weather", 23)
                Call ComputeMeasureValue(i, j, "Temperature", 24)
                Call ComputeMeasureValue(i, j, "Urban Areas", 25)
                Call ComputeMeasureValue(i, j, "Terrain", 26)
                Application.ScreenUpdating = False
                'WEIGHT THE VALUE SCORES
                'Multiply Measures by Global Weights stored on Weights Sheet B106:AB106

```

```

WRow = 106
For wt = 0 To 26
Value = ValueSheet.Cells(j, VMCOL + wt).Value
Weight = WeightSheet.Cells(WRow, WCOL + wt).Value
WeightedVal = Value * Weight
ValueSheet.Cells(j, VMCOL + wt) = WeightedVal
Next wt
'Calculate TOTAL Value by copying formula from C3
Application.ScreenUpdating = False
AltTotal = "C" & Application.WorksheetFunction.Text(j, 0)
ValueSheet.Range("C4").Copy ValueSheet.Range(AltTotal)
'Apply scenario factors
'First, Get Measures
Probability = ODSheet.Cells(FirstOD + dest - 1, 12)
NationalSecurityMin = ODSheet.Cells(FirstOD + dest - 1, 13)
NationalSecurityMode = ODSheet.Cells(FirstOD + dest - 1, 14)
NationalSecurityMax = ODSheet.Cells(FirstOD + dest - 1, 15)
'Next Get Weighting
ValueWt = WeightSheet.Cells(75, 2)
ProbabilityWt = WeightSheet.Cells(75, 3)
NationalSecurityWt = WeightSheet.Cells(75, 4)
'Next Get Weighted Scores
ValueScore = ValueSheet.Cells(j, 3) * ValueWt
'Get the prospect value that is computed when destination is entered
ProspectValue = ODSheet.Cells(FirstOD + dest - 1, 30)
ProbabilityScore = ProspectValue * ProbabilityWt
'For national security, an expected value is taken for the triangle
distribution input
s
NationalSecurityScore = ((Get_Cont_Values("National Security",
NationalSecurityMin) +
Get_Cont_Values("National Security", NationalSecurityMode)
+
Get_Cont_Values("National Security", NationalSecurityMax))
/ 3)
* NationalSecurityWt
Score = ValueScore + ProbabilityScore + NationalSecurityScore
EnrouteName = ValueSheet.Cells(j, 2)
'WRITE SCORE TO SCENARIO MATRIX
'First find the correct alternative
Search = 5
DestinationScenarios - 3
While ScoreSheet.Cells(Search, 2) <> EnrouteName
Search = Search + 1
Wend
'Write the score in the matrix and change formatting to show a valid
alternative score
within the lens
ScoreSheet.Unprotect
ScoreSheet.Activate
ScoreSheet.Cells(Search, dest + 3) = Score
ScoreSheet.Cells(Search, dest + 3).NumberFormat = "0.000"
ScoreCell = ColumnChr(dest + 3) & Application.WorksheetFunction.Text(Search,
0)
Range(ScoreCell).Select
Selection.Font.Italic = False
Selection.Font.ColorIndex = xlAutomatic
Else 'If alternative is not in lens, format the 0 score to indicate not in
lens
'WRITE SCORE TO SCENARIO MATRIX
'First find the correct alternative
EnrouteName = DataSheet.Cells(Alternative + 1, 1)
Search = 5

```

```

While ScoreSheet.Cells(Search, 2) <> EnrouteName
Search = Search + 1
Wend
ScoreSheet.Unprotect
ScoreSheet.Activate
' Since not in lens, format the zero score with color to highlight
ScoreCell = ColumnChr(dest + 3) & Application.WorksheetFunction.Text(Search,
0)
Range(ScoreCell).Select
Selection.Font.Italic = False
Selection.Font.ColorIndex = 9
End If
Alternative = Alternative + 1
Wend
Else ' if no enroute required, change the formatting of the Destination name
ScoreSheet.Unprotect
ScoreSheet.Activate
'Cells(4, Dest + 3).Select
DestCell = ColumnChr(dest + 3) & "4"
Range(DestCell).Select
Selection.Interior.ColorIndex = 2
Selection.Font.ColorIndex = 16
Selection.Font.Italic = True
End If
Next dest
'Sort Scenario Matrix
NumAlt = Application.WorksheetFunction.CountA(DataSheet.Range("A2:A65356"))
SortRange = "B4:" & ColumnChr(64 + NumDest + 3) &
Application.WorksheetFunction.Text(4 + NumAlt, 0)
)
ScoreSheet.Activate
ScoreSheet.Range(SortRange).Select
If ScoreSheet.Cells(2, 2) <> 0 Then
Selection.Sort Key1:=Range("C5"), Order1:=xlDescending, Key2:=Range("B5") _ ,
Order2:=xlAscending, Header:=xlGuess, OrderCustom:=1, MatchCase:=
False, Orientation:=xlTopToBottom, DataOption1:=xlSortNormal, DataOption2 _ :=
xlSortNormal
End If
'Write the rank number for the ranked alternatives
ScoreSheet.Unprotect
For i = 1 To NumAlt
Cells(i + 4, 1) = i
Next i
ODSheet.Cells(2, 2) = OldDestIndex
ValueSheet.Protect
DestinationScenarios - 4
DataSheet.Protect
ScoreSheet.Protect
WeightSheet.Protect
Application.ScreenUpdating = True
End Sub
Sub InitializeMatrix()
' Zeroes out the initial scenario matrix. Writes the origin names. Writes all
destinations along col
umn headers.
' Writes all alternatives names at the beginning of row. Sets all matrix
values equal to zero.
' All scores are initially formatted in italics and with color to signify that
no enroute is required
' When it is determined that an enroute is required in ComputeScenario, the
formatting is changed.
Set ODSheet = Worksheets("Orig-Dest Airfields")
Set ScoreSheet = Worksheets("Scenario Matrix")
Set DataSheet = Worksheets("Airfields Data")

```

```

ScoreSheet.Unprotect
ODSheet.Unprotect
DataSheet.Unprotect
Application.ScreenUpdating = False
ScoresDisplayed = False
OriginIndex = ODSheet.Cells(1, 2)
'Delete values from previously scored matrix
ScoreSheet.Rows("5:65356").Delete
ScoreSheet.Range("D4:IV4").Delete
Rows("4:4").Select
'Reset color formatting
With Selection.Interior
.ColorIndex = 2
.Pattern = xlSolid
End With
Selection.Font.ColorIndex = 0
'Write the origin name in the Score Matrix
ScoreSheet.Cells(1, 4) = ODSheet.Cells(FirstOD + OriginIndex - 1, 1)
'Write Destinations in Each Heading
NumDest = Application.WorksheetFunction.CountA(ODSheet.Range("K11:K65356"))
For dest = 1 To NumDest
ScoreSheet.Cells(4, dest + 3) = ODSheet.Cells(FirstOD + dest - 1, 11)
Next dest
'Write Alternatives' Names in Rows
NumAlt = Application.WorksheetFunction.CountA(DataSheet.Range("A2:A65356"))
For Alt = 1 To NumAlt
ScoreSheet.Cells(Alt + 4, 2) = DataSheet.Cells(Alt + 1, 1)
RowRange = "D" & Application.WorksheetFunction.Text(Alt + 4, 0) & ":IV" & _
Application.WorksheetFunction.Text(Alt + 4, 0)
ScoreSheet.Cells(Alt + 4, 3) = "=Sum(" & RowRange & ")"
Next Alt
' Initialize Matrix with all zeroes written in italics and with font color to
signify no enroute r
equired
For dest = 1 To NumDest
For Alt = 1 To NumAlt
ScoreSheet.Cells(Alt + 4, dest + 3) = 0
ScoreCell = ColumnChr(dest + 3) & Application.WorksheetFunction.Text(Alt + 4,
0)
Range(ScoreCell).Select
Selection.Font.Italic = True
Selection.Font.ColorIndex = 16
Next Alt
Next dest
ScoreSheet.Protect
DestinationScenarios - 5
ODSheet.Protect
DataSheet.Protect
' Application.ScreenUpdating = True
End Sub
Sub ScenarioRankings()
' Creates a bar chart of the top 10 ranking alternatives
Set ScoreSheet = Worksheets("Scenario Matrix")
Set RankSheet = Worksheets("Scenario Rankings")
Set ODSheet = Worksheets("Orig-Dest Airfields")
Set DataSheet = Worksheets("Airfields Data")
ScoreSheet.Unprotect
RankSheet.Unprotect
Application.ScreenUpdating = False
OriginIndex = ODSheet.Cells(1, 2)
ScoreSheet.Activate
'Write the scores next to Enroute Airfield Names if it has not already been
done
If (Not (ScoresDisplayed)) Then

```

```

NumAlt = Application.WorksheetFunction.CountA(DataSheet.Range("A2:A65356"))
For i = 1 To NumAlt
Cells(i + 4, 2) = Cells(i + 4, 2) & " " & Application.WorksheetFunction.Text _
(Cells(i + 4, 3), "0.000")
Next i
Columns("B").AutoFit
ScoresDisplayed = True
End If
' -----CREATE RANKINGS CHART-----
-----
RankSheet.Activate
RankSheet.Range("L9") = ODSheet.Cells(FirstOD + OriginIndex - 1, 1)
ActiveSheet.ChartObjects.Delete 'Delete Previous Sheet
Set Rank = RankSheet.ChartObjects.Add(5, 5, 475, 385)
Rank.Name = "Rank"
With RankSheet.DrawingObjects("Rank")
.Placement = xlFreeFloating
.PrintObject = True
End With
RankSheet.ChartObjects("Rank").Activate
RankSheet.ChartObjects("Rank").Name = "Rank"
ActiveChart.ChartType = xlBarStacked
NumDest = Application.WorksheetFunction.CountA(ODSheet.Range("K11:K65356"))
ChartRange = "B4:B14,D4:" & ColumnChr(NumDest + 3) & "14"
ActiveChart.SetSourceData Source:=Sheets("Scenario Matrix").Range(ChartRange),
PlotBy:=xlColumns
' Chart and Plot area formatting
ActiveChart.Location Where:=xlLocationAsObject, Name:="Scenario Rankings"
ActiveChart.Axes(xlCategory, xlPrimary).CategoryType = xlAutomatic
With ActiveChart.Axes(xlCategory)
.HasMajorGridlines = False
.HasMinorGridlines = False
End With
With ActiveChart.Axes(xlValue)
.HasMajorGridlines = False
.HasMinorGridlines = False
End With
With ActiveChart
.HasTitle = True
.ChartTitle.Characters.Text = "Top 10 EnRoute Airfields"
End With
DestinationScenarios = 6
ActiveChart.Axes(xlValue).Select
Selection.Delete
ActiveChart.HasLegend = True
ActiveChart.Legend.Select
Selection.Position = xlBottom
ActiveChart.PlotArea.Select
With Selection.Border
.LineStyle = xlNone
End With
With Selection.Interior
.ColorIndex = 2
.PatternColorIndex = 1
.Pattern = xlSolid
End With
ActiveChart.Axes(xlCategory).Select
With Selection.Border
.LineStyle = xlNone
End With
With Selection
.MajorTickMark = xlOutside
.MinorTickMark = xlNone

```

```

.TickLabelPosition = xlNextToAxis
End With
With ActiveChart.Axes(xlCategory)
.CrossesAt = 1
.TickLabelSpacing = 1
.TickMarkSpacing = 1
.AxisBetweenCategories = True
.ReversePlotOrder = True
End With
ActiveChart.ChartArea.Select
With Selection.Border
.Weight = 2
.LineStyle = 0
End With
Selection.Interior.ColorIndex = xlAutomatic
ActiveChart.PlotArea.Select
'Spacing between bars
With ActiveChart.ChartGroups(1)
.Overlap = 100
.GapWidth = 50
.HasSeriesLines = False
End With
'Placing the legend and ensuring the font size on the axis will always be
legible
ActiveChart.Legend.Select
Selection.Left = 33
Selection.Width = 414
Selection.Position = xlBottom
Selection.Height = 99
Selection.Top = 278
ActiveChart.PlotArea.Select
Selection.Width = 452
Selection.Height = 222
ActiveChart.ChartArea.Select
ActiveChart.Axes(xlCategory).Select
With Selection.TickLabels.Font
.Size = 8
End With
-----END RANKINGS CHART-----
-----
RankSheet.Visible = True
Range("O1").Select
RankSheet.Protect
ScoreSheet.Protect
Application.ScreenUpdating = True
End Sub
Sub InitializeSensitivity(CurrentWt)
' Prepares the Scenario Sensitivity Chart for new data
DestinationScenarios = 7
' Data on this sheet is stored beginning in Row 5000
' Must pass in the CurrentWt of the objective of interest
Set ChartSheet = Worksheets("Scenario Sensitivity")
Set DataSheet = Worksheets("Airfields Data")
DataSheet.Unprotect
ChartSheet.Unprotect
Application.ScreenUpdating = False
ChartSheetActivate
'Delete previous chart data
ChartSheet.Rows("5001:65356").Delete
'Draw a line for the current weight
ChartSheet.Cells(5000, 4) = CurrentWt - 0.0001
ChartSheet.Cells(5000, 5) = CurrentWt + 0.0001
Count = Application.WorksheetFunction.CountA(DataSheet.Range("A2:A65356"))

```

```

'Put the names of the enroute alternatives in according to the order of the
current data sheet
For i = 1 To Count
ChartSheet.Cells(5000 + i, 1) = DataSheet.Cells(i + 1, 1)
Next i
DataSheet.Protect
ChartSheet.Protect
Application.ScreenUpdating = True
End Sub
Sub ComputeScenarioSensitivity(SensitivityRun)
' Called by the ScenarioMenu UserForm. SensitivityRun determines if the values
being determined are fo
r
' when the weight of interest equals zero or one.
' Computes the sensitivity analysis values for the operational value hierarchy
Set DataSheet = Worksheets("Airfields Data")
Set ScoreSheet = Worksheets("Scenario Matrix")
Set ChartSheet = Worksheets("Scenario Sensitivity")
DataSheet.Unprotect
ScoreSheet.Unprotect
ChartSheet.Unprotect
ScoreSheet.Activate
Application.ScreenUpdating = False
Call ComputeScenario 'Compute values with given weights
NumAlt = Application.WorksheetFunction.CountA(DataSheet.Range("A2:A65356"))
' Search the Chart Sheet for the correct row of the alternative, then write
the result for the Sen
sitivity Run
For Alt = 1 To NumAlt
EnrouteName = ScoreSheet.Cells(Alt + 4, 2)
Search = 5001
While ChartSheet.Cells(Search, 1) <> EnrouteName
Search = Search + 1
Wend
ChartSheet.Cells(Search, SensitivityRun + 2) = ScoreSheet.Cells(Alt + 4, 3)
Next Alt
' Enter data to draw a line for the current weight
ChartSheet.Cells(5000 + NumAlt + 1, 1) = "Current Weight"
ChartSheet.Cells(5000 + NumAlt + 1, 2) = ""
ChartSheet.Cells(5000 + NumAlt + 1, 3) = ""
ChartSheet.Cells(5000 + NumAlt + 1, 4) = -999
DestinationScenarios = 8
ChartSheet.Cells(5000 + NumAlt + 1, 5) = 1000
DataSheet.Protect
ScoreSheet.Protect
ChartSheet.Protect
'Application.ScreenUpdating = True
End Sub
Sub DrawScenarioSensitivity(Measure)
' Draws the Scenario Sensitivity Chart for ALL alternatives
Set DataSheet = Worksheets("Airfields Data")
Application.ScreenUpdating = False
'===== DRAW THE SENSITIVITY CHART =====
sheetname = "Scenario Sensitivity"
Set SensSheet = Worksheets(sheetname)
Set ODSheet = Worksheets("Orig-Dest Airfields")
SensSheetActivate
SensSheet.Unprotect
NumAlt = Application.WorksheetFunction.CountA(DataSheet.Range("A2:A65356"))
OriginIndex = ODSheet.Cells(1, 2)
' Label the chart with the selected origin
SensSheet.Range("L12") = ODSheet.Cells(FirstOD + OriginIndex - 1, 1)
ActiveSheet.ChartObjects.Delete 'Delete Previous Sheet

```

```

Set ch = SensSheet.ChartObjects.Add(5, 5, 475, 365)
ch.Name = "Chart1"
With SensSheet.DrawingObjects("Chart1")
    .Placement = xlFreeFloating
    .PrintObject = True
End With
SensSheet.ChartObjects("Chart1").Activate
SensSheet.ChartObjects("Chart1").Name = "Chart1"
ActiveChart.ChartType = xlXYScatterSmoothNoMarkers
ActiveChart.Location Where:=xlLocationAsObject, Name:=sheetname
ActiveChart.SeriesCollection.NewSeries
WtStr = Application.WorksheetFunction.Text(SensSheet.Cells(5000, 4) + 0.0001,
"0.000")
TitleStr = SensType & " Scenario Sensitivity Analysis on " & vbCrLf & Measure
& vbCrLf &
"Current weight is " & WtStr
With ActiveChart
    .HasTitle = True
    .Axes(xlCategory, xlPrimary).HasTitle = False
    .Axes(xlValue, xlPrimary).HasTitle = True
    .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Value"
    .ChartTitle.Characters.Text = TitleStr
End With
ActiveChart.HasLegend = True
'Fix value axis
MaxStr = "=Int(Max(B5001:C" & Application.WorksheetFunction.Text(5000 + NumAlt
+ 1, 0) & "))+1
"
SensSheet.Cells(4999, 7) = MaxStr
MaxScore = SensSheet.Cells(4999, 7)
With ActiveChart.Axes(xlValue)
    .MinimumScale = 0
    .MaximumScale = MaxScore
    .MinorUnitIsAuto = True
    .MajorUnitIsAuto = True
    .Crosses = xlAutomatic
    .ReversePlotOrder = False
    .ScaleType = xlLinear
    .DisplayUnit = xlNone
    DestinationScenarios - 9
End With
ActiveChart.Axes(xlValue).MajorGridlines.Select
With Selection.Border
    .ColorIndex = 57
    .Weight = xlHairline
    .LineStyle = xlDot
End With
With ActiveChart.Axes(xlCategory)
    .MinimumScale = 0
    .MaximumScale = 1
    .MinorUnitIsAuto = True
    .MajorUnit = 0.1
    .Crosses = xlAutomatic
    .ReversePlotOrder = False
    .ScaleType = xlLinear
    .DisplayUnit = xlNone
End With
SensSheet.DrawingObjects("Chart1").RoundedCorners = True
SensSheet.DrawingObjects("Chart1").Shadow = True
DataRange = "A5000:E" & Application.WorksheetFunction.Text(5000 + NumAlt + 1,
0)
ActiveChart.SetSourceData Source:=Sheets(sheetname).Range(DataRange), _
PlotBy:=xlRows
' Make Lines Thick

```

```

For i = 1 To NumAlt
ActiveChart.SeriesCollection(i).Select
With Selection.Border
.Weight = xlThick
.LineStyle = xlContinuous
End With
Next i
ActiveChart.SeriesCollection(NumAlt + 1).Select
With Selection.Border
.ColorIndex = 3
End With
' Give Chart Area a Grey Background
ActiveChart.ChartArea.Select
Selection.Fill.OneColorGradient Style:=msoGradientHorizontal, Variant:=1, _
Degree:=0.231372549019608
With Selection
.Fill.Visible = True
.Fill.ForeColor.SchemeColor = 15
End With
ActiveChart.Legend.Select
Selection.Font.Size = 7
'Give Plot Area a Blue Background
ActiveChart.PlotArea.Select
Selection.Fill.TwoColorGradient Style:=msoGradientHorizontal, Variant:=2
With Selection
.Fill.Visible = True
.Fill.ForeColor.SchemeColor = 24
.Fill.BackColor.SchemeColor = 17
End With
ActiveChart.ChartTitle.Left = 60
' ===== END OF CHART DRAWING
=====
SensSheet.Protect
Application.ScreenUpdating = True
End Sub
Function ColumnChr(ColumnNum) As String
' Function called to convert the column number to its alpha title
DestinationScenarios - 10
' Without this conversion the maximum number of destinations that could be
written would be 26
' This allows full use of all 256 columns (minus the 3 columns used for rank,
name, and score info)
If ColumnNum < 27 Then
ColumnChr = Chr(64 + ColumnNum)
Else
CC1 = Chr(64 + Int(ColumnNum / 26))
CC2 = Chr(64 + (ColumnNum Mod 26))
ColumnChr = CC1 & CC2
End If
End Function
Sub GoalSeekProspect()
' Allows us to use the inverse of the prospect function to evaluate on the
SDVF sheet
Probability = Cells(23, 13)
ActiveSheet.Unprotect
Range("D100").GoalSeek Goal:=Probability, ChangingCell:=Range("M24")
ActiveSheet.Protect
End Sub

DistanceCalculations - 1
' MODULE FOR DISTANCE CALCULATIONS
Public Const FirstOD = 11 ' First Row for possible Orig-Dest
Public Const DMCOL = 7 ' First Column in DataSheet for Measure

```

```

' The Great Circle distance calculations are performed in the Orig-Dest
Airfields spreadsheet cells B4
to I7
Sub ConvLatLon(NorS, Lat, EorW, Lon)
' Converts Latitude to Negative if S and Longitude to Negative if W
If NorS = "S" Then
Lat = -1 * Lat
End If
If EorW = "W" Then
Lon = -1 * Lon
End If
End Sub
Sub ComputeDistances()
' Utilizes the formulas in the Orig-Dest Airfields spreadsheet
' Calculates the distances from Orig-Alternative, Alternative-Dest, Orig-Dest
' Converts the input Airfield Data lat/long to Decimal-Degree format for
computation
' Computes Critical Leg
' Computes Delta Flight Length
' Determines if each alternative is in the lens
Dim ONorS, OEorW, DNorS, DEorW, ANorS, AEorW As String
Set DataSheet = Worksheets("Airfields Data")
Set ODSheet = Worksheets("Orig-Dest Airfields")
DataSheet.Unprotect
ODSheet.Unprotect
OrigIndex = ODSheet.Cells(1, 2)
DestIndex = ODSheet.Cells(2, 2)
OLat = ODSheet.Cells(OrigIndex + FirstOD - 1, 8)
OLon = ODSheet.Cells(OrigIndex + FirstOD - 1, 9)
ODSheet.Range("B5") = OLat
ODSheet.Range("E5") = OLon
DLat = ODSheet.Cells(DestIndex + FirstOD - 1, 24)
DLon = ODSheet.Cells(DestIndex + FirstOD - 1, 25)
ODSheet.Range("B7") = DLat
ODSheet.Range("E7") = DLon
DistOtoD = ODSheet.Range("I7")
Count = Application.WorksheetFunction.CountA _
(DataSheet.Range("A2:A65356"))
For i = 2 To Count + 1 'For each alternative in the Airfields Data Sheet
'Convert to Decimal-degree format
ANorS = DataSheet.Cells(i, 2)
ALat = DataSheet.Cells(i, 3)
AEorW = DataSheet.Cells(i, 4)
ALon = DataSheet.Cells(i, 5)
Call ConvLatLon(ANorS, ALat, AEorW, ALon)
ODSheet.Range("B6") = ALat
ODSheet.Range("E6") = ALon
DistOtoA = ODSheet.Range("I5")
DistAtoD = ODSheet.Range("I6")
'Write In Lens to DataSheet and Uncertainty Sheet to make sure the Airfields
sort the same
DistanceCalculations = 2
DataSheet.Cells(i, 6) = InLens(DistOtoA, DistAtoD)
Set USheet = Worksheets("Uncertainty Data")
USheet.Unprotect
USheet.Cells(i + 2, 2) = DataSheet.Cells(i, 6)
USheet.Protect
'Determine Critical Leg
DataSheet.Cells(i, DMCOL + 1) = CriticalLeg(DistOtoA, DistAtoD)
'Determine Delta Flight Length if in the leg, else set Delta to 9999, i.e.
penalize alternativ
e
If InLens(DistOtoA, DistAtoD) Then
DataSheet.Cells(i, DMCOL + 2) = DeltaFlight(DistOtoA, DistAtoD, DistOtoD)

```

```

Else: DataSheet.Cells(i, DMCol + 2) = 9999
End If
DataSheet.Cells(i, DMCol + 1).NumberFormat = "0"
DataSheet.Cells(i, DMCol + 2).NumberFormat = "0"
Next i
Call SortFields
DataSheet.Protect
ODSheet.Protect
End Sub
Function CriticalLeg(DistOtoA, DistAtoD)
If DistOtoA >= DistAtoD Then
CriticalLeg = DistOtoA
Else
CriticalLeg = DistAtoD
End If
End Function
Function DeltaFlight(DistOtoA, DistAtoD, DistOtoD)
DeltaFlight = (DistOtoA + DistAtoD) - DistOtoD
End Function
Function InLens(DistOtoA, DistAtoD) As Boolean
' Determines if alternative is in the lens by comparing its distances to the
maximum critical leg dist
ance
Call GetContVFRRange("Critical Leg", lo, hi)
InLens = (DistOtoA <= hi) And (DistAtoD <= hi)
End Function
Function EnrouteRequired(DistOtoD) As Boolean
Call GetContVFRRange("Critical Leg", lo, hi)
EnrouteRequired = (DistOtoD > hi)
End Function

HierarchyForms = 1
' VALUE HIERARCHY WEIGHTS USERFORMS
' Assigns macros to show appropriate Userform when the clicking on the value
hierarchy
Sub ThroughputMenu()
Throughput.Show
End Sub
Sub EnrouteDiversionMenu()
EnrouteDiversion.Show
End Sub
Sub SustainmentMenu()
Sustainment.Show
End Sub
Sub BaseMenu()
Base.Show
End Sub
Sub FuelMenu()
Fuel.Show
End Sub
Sub BaseInfrastructureMenu()
BaseInfrastructure.Show
End Sub
Sub GeographyMenu()
Geography.Show
End Sub
Sub FacilitiesMenu()
Facilities.Show
End Sub
Sub HostNationRelationsMenu()
HostNationRelations.Show
End Sub
Sub GroundTransportationMenu()
GroundTransportation.Show

```

```

End Sub
Sub UtilitiesMenu()
    Utilities.Show
End Sub
Sub AircraftMenu()
    Aircraft.Show
End Sub
Sub EnrouteScoreMenu()
    EnrouteScore.Show
End Sub
' ROUTINES TO CALL UP MEASURE VALUE FUNCTIONS WHEN CLICKED
' It is IMPORTANT to use the CORRECT sheet name when referring to each measure
Sub ShowVFHelp()
    VFHelp.Show Modeless
End Sub
Sub ShowVF(MeasureName As String)
    ' Shows the current Value Function and hides the rest
    ' Should never run into the Nosheet scenario
    ' This is left over from Lt Col Weir's generic hierarchy builder and kept for
    error checking
    Application.ScreenUpdating = False
    Count = 1
    While (Sheets(Count).Name <> MeasureName)
        Count = Count + 1
    If Count > Sheets.Count Then GoTo Nosheet
    Wend
    Nosheet:
    If Count > Sheets.Count Then
        MsgBox "A Value Function Sheet for " & MeasureName & " does not exist"
    Else
        For i = Sheets.Count To 1 Step -1
            If Sheets(i).Name = MeasureName Then
                Sheets(i).Visible = True
            Else
                Sheets(i).Visible = False
            End If
        Next i
        HierarchyForms = 2
    End If
    Application.ScreenUpdating = True
End Sub
Sub MOGVF()
    Call ShowVF("MOG")
End Sub
Sub CriticalLegVF()
    Call ShowVF("Critical Leg")
End Sub
Sub FlightLengthVF()
    Call ShowVF("Delta Flight Length")
End Sub
Sub AltAirfieldsVF()
    Call ShowVF("Alternate Airfields")
End Sub
Sub FuelStorageVF()
    Call ShowVF("Fuel Storage")
End Sub
Sub FuelResupplyVF()
    Call ShowVF("Fuel Resupply")
End Sub
Sub DipClearanceVF()
    Call ShowVF("Diplomatic Clearance")
End Sub
Sub ForceProtectionsVF()
    Call ShowVF("Force Protection")

```

```

End Sub
Sub DepStateVF()
Call ShowVF("Dept of State")
End Sub
Sub MilCoopVF()
Call ShowVF("Military Cooperation")
End Sub
Sub SeaportVF()
Call ShowVF("Seaport")
End Sub
Sub RailVF()
Call ShowVF("Railroad")
End Sub
Sub RoadVF()
Call ShowVF("Road System")
End Sub
Sub CommAirportVF()
Call ShowVF("Commercial Airport")
End Sub
Sub LodgingVF()
Call ShowVF("Lodging")
End Sub
Sub DiningVF()
Call ShowVF("Dining")
End Sub
Sub MedicalVF()
Call ShowVF("Medical")
End Sub
Sub SewerVF()
Call ShowVF("Sewer")
End Sub
Sub PowerVF()
Call ShowVF("Power")
End Sub
Sub WaterVF()
Call ShowVF("Potable Water")
End Sub
Sub CommVF()
Call ShowVF("Communications")
End Sub
Sub MountainousVF()
Call ShowVF("Mountainous")
End Sub
Sub AltitudeVF()
Call ShowVF("Altitude")
End Sub
Sub TempVF()
HierarchyForms - 3
Call ShowVF("Temperature")
End Sub
Sub WxVF()
Call ShowVF("Weather")
End Sub
Sub UrbanAreasVF()
Call ShowVF("Urban Areas")
End Sub
Sub TerrainVF()
Call ShowVF("Terrain")
End Sub

MenuNavigation - 1
' ROUTINES FOR NAVIGATING THE MANY MENUS
Sub GerbilClick()
Worksheets("Main").Shapes("Gerbil Quote").Visible = _

```

```

Not (Worksheets("Main").Shapes("Gerbil Quote").Visible)
End Sub
Sub ShowAllSheets()
' CTRL-SHIFT-A for shortcut ket
' For programming purposes
' Shown on Main Menu sheet
' Even though all sheets are displayed and unprotected, running any macro
attached to a button may
' hide a sheet and protect another sheet
Application.ScreenUpdating = False
For i = 1 To Sheets.Count
Sheets(i).Visible = True
Next i
Call UnprotectProgram
Sheet1.Activate
Application.ScreenUpdating = True
End Sub
Sub ShowHelp()
MainHelp.Show
End Sub
Sub ShowRankChart()
Application.ScreenUpdating = False
Call CreateRankings(1)
Worksheets("Rankings").Visible = True
Worksheets("Main").Visible = False
Worksheets("Values").Visible = False
Worksheets("Rankings").Activate
With ActiveWindow
.DisplayHeadings = False
.DisplayOutline = False
.DisplayHorizontalScrollBar = False
.DisplayVerticalScrollBar = False
End With
ActiveSheet.ScrollArea = "A1: Q31" 'limit ability to scroll
Application.ScreenUpdating = True
End Sub
Sub ReturnToHierarchy()
Application.ScreenUpdating = False
Worksheets("Hierarchy").Visible = True
ActiveSheet.Visible = False
Worksheets("Hierarchy").Activate
Application.ScreenUpdating = True
End Sub
Sub ReturnToScenarioHierarchy()
Application.ScreenUpdating = False
Worksheets("Main").Visible = True
ActiveSheet.Visible = False
Worksheets("Main").Activate
Application.ScreenUpdating = True
ScenarioMenu.Show
End Sub
Sub GoToNationalSecurityDefinitions()
Application.ScreenUpdating = False
Worksheets("National Security Definitions").Visible = True
ActiveSheet.Visible = False
Worksheets("National Security Definitions").Activate
Application.ScreenUpdating = True
End Sub
Sub ReturnToOrigDest()
Application.ScreenUpdating = False
Worksheets("Orig-Dest Airfields").Visible = True
ActiveSheet.Visible = False
Worksheets("Orig-Dest Airfields").Activate
Application.ScreenUpdating = True

```

```

End Sub
Sub ShowHierarchyHelp()
HierarchyHelp.Show
MenuNavigation - 2
End Sub
Sub ShowScores()
Application.ScreenUpdating = False
Worksheets("Values").Visible = True
Worksheets("Values").Activate
' ActiveSheet.Shapes("Markings").Visible = True
Worksheets("Main").Visible = False
Worksheets("Rankings").Visible = False
With ActiveWindow
.DisplayHorizontalScrollBar = True
.DisplayVerticalScrollBar = True
.ScrollColumn = 1
.ScrollRow = 1
End With
ActiveSheet.ScrollArea = ""
Application.ScreenUpdating = True
End Sub
Sub ShowScenarioMenu()
ScenarioMenu.Show
End Sub
Sub ShowDefinitions()
Definitions.Show
End Sub
Sub ShowScenarioMatrix()
Application.ScreenUpdating = False
If ActiveSheet.Name <> "Scenario Rankings" Then
Call ComputeScenario
End If
Worksheets("Scenario Matrix").Visible = True
Worksheets("Scenario Matrix").Activate
Worksheets("Main").Visible = False
Worksheets("Scenario Rankings").Visible = False
With ActiveWindow
.DisplayHorizontalScrollBar = True
.DisplayVerticalScrollBar = True
.ScrollColumn = 1
.ScrollRow = 1
End With
ActiveSheet.ScrollArea = ""
Application.ScreenUpdating = True
End Sub
Sub ShowScenarioRankings()
Application.ScreenUpdating = False
If ActiveSheet.Name <> "Scenario Matrix" Then
Call ComputeScenario
End If
Call ScenarioRankings
Worksheets("Scenario Rankings").Visible = True
Worksheets("Scenario Rankings").Activate
Worksheets("Main").Visible = False
Worksheets("Scenario Matrix").Visible = False
Application.ScreenUpdating = True
End Sub
Sub ShowScenarioSensitivity()
Application.ScreenUpdating = False
Worksheets("Scenario Sensitivity").Visible = True
Worksheets("Scenario Sensitivity").Activate
Worksheets("Main").Visible = False
ActiveSheet.PageSetup.PrintArea = "$A$1:$L$31"
ActiveSheet.DisplayAutomaticPageBreaks = False

```

```

Application.ScreenUpdating = True
End Sub
Sub ShowOrigDest()
Application.ScreenUpdating = False
Worksheets("Orig-Dest Airfields").Visible = True
ActiveSheet.Visible = False
Worksheets("Orig-Dest Airfields").Activate
ActiveSheet.Unprotect
Rows("1:8").Select
' Hide the top 8 rows which contain the cells that capture the index number
from the Main menu dro
p down lists
' and the Great Circle distance calculations
MenuNavigation - 3
Selection.EntireRow.Hidden = True
Range("A1").Select
ActiveSheet.Protect
Application.ScreenUpdating = True
End Sub
Sub ReturnToMain()
Application.ScreenUpdating = False
Worksheets("Main").Visible = True
ActiveSheet.Visible = False
Worksheets("Main").Activate
With ActiveWindow
.DisplayHorizontalScrollBar = False
.DisplayVerticalScrollBar = False
End With
ActiveSheet.Protect
Application.ScreenUpdating = True
End Sub
Sub ShowAlternatives()
Application.ScreenUpdating = False
' Computes the Distances for when new Orig-Dest pairs are selected
Call ComputeDistances
Set ODSheet = Worksheets("Orig-Dest Airfields")
DistOtoD = ODSheet.Range("I7")
OrigIndex = ODSheet.Cells(1, 2)
DestIndex = ODSheet.Cells(2, 2)
If Not (EnrouteRequired(DistOtoD)) Then
Orig = ODSheet.Cells(OrigIndex + FirstOD - 1, 1)
dest = ODSheet.Cells(DestIndex + FirstOD - 1, 1)
DistStr = "The great circle distance from " & Orig & " to " & dest & " is only
" &
Round(DistOtoD, 0) & " nautical miles"
Msg = DistStr & vbCrLf & vbCrLf & "Do you wish to continue?"
Ans = MsgBox(Msg, vbYesNo, "NOTE: An en route stop is not required")
If (Ans = vbNo) Then GoTo StayPut:
End If
Worksheets("Airfields Data").Visible = True
Worksheets("Main").Visible = False
Worksheets("Airfields Data").Activate
StayPut:
Application.ScreenUpdating = True
End Sub
Sub ShowRunModel()
Application.ScreenUpdating = False
' Computes the Distances for when new Orig-Dest pairs are selected
Call ComputeDistances
Set ODSheet = Worksheets("Orig-Dest Airfields")
DistOtoD = ODSheet.Range("I7")
OrigIndex = ODSheet.Cells(1, 2)
DestIndex = ODSheet.Cells(2, 2)
If Not (EnrouteRequired(DistOtoD)) Then

```

```

Orig = ODSheet.Cells(OrigIndex + FirstOD - 1, 1)
dest = ODSheet.Cells(DestIndex + FirstOD - 1, 11)
DistStr = "The great circle distance from " & Orig & " to " & dest & " is only
" &
Round(DistOtoD, 0) & " nautical miles"
Msg = DistStr & vbCrLf & vbCrLf & "Do you wish to continue?"
Ans = MsgBox(Msg, vbYesNo, "NOTE: An en route stop is not required")
If (Ans = vbNo) Then GoTo StayPut:
End If
Application.ScreenUpdating = True
RunModelMenu.Show
StayPut:
Application.ScreenUpdating = True
End Sub
MenuNavigation - 4
Sub ShowGlobalChart()
' For click button that goes from Value Hierarchy to Show Global Weights Chart
Application.ScreenUpdating = False
Charts("Global Weights Chart").Visible = True
Worksheets("Hierarchy").Visible = False
Charts("Global Weights Chart").Activate
Application.ScreenUpdating = True
End Sub
Sub ShowWeights()
' For click button that goes from Value Hierarchy to Weights
Application.ScreenUpdating = False
Worksheets("Weights").Visible = True
Worksheets("Hierarchy").Visible = False
Worksheets("Weights").Activate
ActiveSheet.PageSetup.PrintArea = "$A$1:$J$23"
Range("B5").Select
Application.ScreenUpdating = True
End Sub
Sub ShowTacSensHelp()
TacSensitivityHelp.Show
End Sub
Sub ShowOpSensHelp()
OpSensitivityHelp.Show
End Sub
Sub ShowAllWeightsHelp()
AllWeightsHelp.Show
End Sub
Sub NotYetFunctioning()
MsgBox "This Button Not Yet Functional", vbOKOnly, "Construction in Progress"
End Sub
Sub UnprotectProgram()
' Unprotects program for programming and returns you to your current sheet
' Shortcut CTRL-SHIFT-U
Application.ScreenUpdating = False
Set CurrentSheet = ActiveSheet
For i = 1 To Sheets.Count
Sheets(i).Activate
ActiveSheet.Unprotect
If ActiveSheet.Type = xlWorksheet Then 'don't change the charts
ActiveSheet.ScrollArea = "" 'Return ability to scroll
With ActiveWindow 'return the scroll bars and headings
.DisplayHeadings = True
.DisplayOutline = True
.DisplayHorizontalScrollBar = True
.DisplayVerticalScrollBar = True
End With
End If
Next i
With Application

```

```

.ShowStartupDialog = True
.DisplayFormulaBar = True
.ShowWindowsInTaskbar = True
End With
Worksheets("Orig-Dest Airfields").Activate
Rows("1:8").Select
Selection.EntireRow.Hidden = False
Range("A1").Select
CurrentSheet.Activate
ActiveWindow.DisplayHorizontalScrollBar = True
ActiveWindow.DisplayVerticalScrollBar = True
Application.ScreenUpdating = True
End Sub
Sub ApplicationInterface()
' Changes to an Application Interface
MenuNavigation - 5
' Removes Scroll bars and Row/Column Headings
' Protects all worksheets
' Brings you to the Main Menu
' Shortcut CTRL-SHIFT-P
Application.ScreenUpdating = False
Current = ActiveSheet.Name
If Current <> "Main" Then
Sheet1.Visible = True
Sheet1.Activate
Worksheets(Current).Visible = False
End If
For i = 1 To Sheets.Count
Sheets(i).Activate
If ActiveSheet.Type = xlWorksheet Then
With ActiveWindow
.DisplayHeadings = False
.DisplayOutline = False
.DisplayHorizontalScrollBar = False
.DisplayVerticalScrollBar = False
End With
ActiveSheet.ScrollArea = "A1: Q31" 'limit ability to scroll
ActiveSheet.Protect
ActiveSheet.EnableSelection = xlUnlockedCells
ActiveSheet.DisplayAutomaticPageBreaks = False
Else: ActiveSheet.Protect
End If
If i <> 1 Then ActiveSheet.Visible = False
Next i
Call ProtectMeasures
With Application
>ShowStartupDialog = False
.DisplayFormulaBar = False
>ShowWindowsInTaskbar = False
End With
Sheet1.Activate
Application.ScreenUpdating = True
End Sub

SendToFile - 1
' This module allows the results of the program file to be saved to another
file location
' This allows the user to save and edit the formatting of the results for
presentation
Function FileExists(fname) As Boolean
FileExists = (Dir(fname) <> "")
End Function
Function SheetExists(sname) As Boolean
Dim x As Object

```

```

On Error Resume Next
Set x = ActiveWorkbook.Sheets(sname)
If Err = 0 Then SheetExists = True _
Else SheetExists = False
End Function
Function WorkbookOpen(wbname) As Boolean
Dim x As Workbook
On Error Resume Next
Set x = Workbooks(wbname)
If Err = 0 Then WorkbookOpen = True _
Else WorkbookOpen = False
End Function
Sub WriteSensitivityToFile()
' Writes sensitivity analysis results to a file in MyDocuments by copying the
sheet and changing the color format
MsgBox "Sheet will be saved to GERBIL MODEL RESULTS.XLS in MyDocuments",
vboKOnly, "Saving Sensitivity Results"
Application.ScreenUpdating = False
Set Program = ActiveWorkbook
If Not FileExists("GERBIL Model Results.xls") Then
If Not WorkbookOpen("GERBIL Model Results.xls") Then
Set NewBook = Workbooks.Add
With NewBook
.Title = "GERBIL Model Results"
.SaveAs Filename:="GERBIL Model Results.xls"
End With
End If
Else
If Not WorkbookOpen("GERBIL Model Results.xls") Then
Workbooks.Open "GERBIL Model Results.xls"
Else
Workbooks("GERBIL Model Results.xls").Activate
End If
End If
'Copy the sheet
Program.Sheets("Sensitivity").Copy After:=Workbooks("GERBIL Model
Results.xls").Sheets(Sheets.Count)
'In results file, delete GERBIL control buttons and text boxes
ActiveSheet.Unprotect
ActiveSheet.Shapes("Button 1").Select
Selection.Delete
ActiveSheet.Shapes("Picture 2").Select
Selection.Delete
ActiveSheet.Shapes("Text Box 3").Select
Selection.Delete
ActiveWindow.DisplayHeadings = True
' Change color formatting of worksheet
Cells.Select
Selection.Interior.ColorIndex = xlNone
Selection.Font.ColorIndex = 0
Range("O1").Select
' Go back to GERBIL
Program.Activate
Application.ScreenUpdating = True
SendToFile - 2
End Sub
Sub WriteScenarioToFile()
' Saves Scenario Matrix and Chart info by copying the Score Matrix to another
file and redrawing a chart in that worksheet

```

```

MsgBox "Sheet will be saved to GERBIL MODEL RESULTS.XLS in MyDocuments",
vboOKOnly, "Saving Scenario
o Results"
Application.ScreenUpdating = False
Set Program = ActiveWorkbook
If Not FileExists("GERBIL Model Results.xls") Then
If Not WorkbookOpen("GERBIL Model Results.xls") Then
Set NewBook = Workbooks.Add
With NewBook
.Title = "GERBIL Model Results"
.SaveAs Filename:="GERBIL Model Results.xls"
End With
End If
Else
If Not WorkbookOpen("GERBIL Model Results.xls") Then
Workbooks.Open "GERBIL Model Results.xls"
Else
Workbooks("GERBIL Model Results.xls").Activate
End If
End If
' Copy Scenario Matrix Sheet and delete unnecessary controls
Program.Sheets("Scenario Matrix").Copy After:=Workbooks("GERBIL Model
Results.xls").Sheets(Sheets.
Count)
ActiveSheet.Unprotect
ActiveSheet.Shapes("Button 1").Select
Selection.Delete
ActiveSheet.Shapes("Button 2").Select
Selection.Delete
ActiveSheet.Shapes("Text Box 3").Select
Selection.Delete
Range("B2") = ""
Range("A1").Select
'CREATING A RANK CHART AS AN OBJECT IN THIS SHEET
Set Rank = ActiveSheet.ChartObjects.Add(50, 225, 475, 385)
Current = ActiveSheet.Name
Rank.Name = "Rank"
With ActiveSheet.DrawingObjects("Rank")
.Placement = xlFreeFloating
.PrintObject = True
End With
ActiveSheet.ChartObjects("Rank").Activate
ActiveSheet.ChartObjects("Rank").Name = "Rank"
ActiveChart.ChartType = xlBarStacked
NumDest = Application.WorksheetFunction.CountA(Program.Worksheets("Orig-Dest
Airfields").Range
("K11:K65356"))
ChartRange = "B4:B14,D4:" & ColumnChr(NumDest + 3) & "14"
ActiveChart.SetSourceData Source:=Sheets(Current).Range(ChartRange), _
PlotBy:=xlColumns
' Chart and Plot area formatting
ActiveChart.Location Where:=xlLocationAsObject, Name:=Current
ActiveChart.Axes(xlCategory, xlPrimary).CategoryType = xlAutomatic
With ActiveChart.Axes(xlCategory)
.HasMajorGridlines = False
.HasMinorGridlines = False
End With
With ActiveChart.Axes(xlValue)
.HasMajorGridlines = False
.HasMinorGridlines = False
End With
With ActiveChart
SendToFile - 3
.HasTitle = True

```

```

.ChartTitle.Characters.Text = "Top 10 EnRoute Airfields"
End With
ActiveChart.Axes(xlValue).Select
Selection.Delete
ActiveChart.HasLegend = True
ActiveChart.Legend.Select
Selection.Position = xlBottom
ActiveChart.PlotArea.Select
With Selection.Border
.LineStyle = xlNone
End With
With Selection.Interior
.ColorIndex = 2
.PatternColorIndex = 1
.Pattern = xlSolid
End With
ActiveChart.Axes(xlCategory).Select
With Selection.Border
.LineStyle = xlNone
End With
With Selection
.MajorTickMark = xlOutside
.MinorTickMark = xlNone
.TickLabelPosition = xlNextToAxis
End With
With ActiveChart.Axes(xlCategory)
.CrossesAt = 1
.TickLabelSpacing = 1
.TickMarkSpacing = 1
.AxisBetweenCategories = True
.ReversePlotOrder = True
End With
ActiveChart.ChartArea.Select
With Selection.Border
.Weight = 2
.LineStyle = 1
End With
Selection.Interior.ColorIndex = xlAutomatic
ActiveChart.PlotArea.Select
'Spacing between bars
With ActiveChart.ChartGroups(1)
.Overlap = 100
.GapWidth = 50
.HasSeriesLines = False
End With
'Resize formatting for the All Measures Chart
If ByTier = 1 Then
ActiveChart.Legend.Select
Selection.Left = 6
Selection.Width = 433
Selection.Height = 88
Selection.Top = 288
ActiveChart.PlotArea.Select
Selection.Width = 426
Selection.Top = 31
Selection.Height = 260
End If
'-----END RANKINGS CHART-----
-----
ActiveSheet.Visible = True
Range("A5").Select
ActiveWindow.FreezePanes = False
Range("A1").Select
'Go back to GERBIL

```

```

Program.Activate
Application.ScreenUpdating = True
End Sub
SendToFile - 4
Sub WriteValuesToFile()
' Saves the Value scores to another file AND draws the Rankings Chart to save
on this worksheet
MsgBox "Sheet will be saved to GERBIL MODEL RESULTS.XLS in MyDocuments",
vboOKOnly, "Saving Value S
core Results"
Application.ScreenUpdating = False
Set Program = ActiveWorkbook
If Not FileExists("GERBIL Model Results.xls") Then
If Not WorkbookOpen("GERBIL Model Results.xls") Then
Set NewBook = Workbooks.Add
With NewBook
.Title = "GERBIL Model Results"
.SaveAs Filename:="GERBIL Model Results.xls"
End With
End If
Else
If Not WorkbookOpen("GERBIL Model Results.xls") Then
Workbooks.Open "GERBIL Model Results.xls"
Else
Workbooks("GERBIL Model Results.xls").Activate
End If
End If
'Copy the Values sheet
Program.Sheets("Values").Copy After:=Workbooks("GERBIL Model
Results.xls").Sheets(Sheets.Count)
'Delete unnecessary info from the program file
ActiveSheet.Unprotect
ActiveSheet.Shapes("Button 1").Select
Selection.Delete
ActiveSheet.Shapes("Button 2").Select
Selection.Delete
ActiveSheet.Shapes("Text Box 4").Select
Selection.Delete
'Copies the current values of the best possible weights cells
' This should remove a link between the Program File and the Results File
ActiveWindow.DisplayHeadings = True
Range("D4:AD4").Select
Selection.Copy
Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _ :=False, Transpose:=False
Range("A1").Select
' Draw a chart by All Measures
Set Rank = ActiveSheet.ChartObjects.Add(300, 300, 475, 385)
Current = ActiveSheet.Name
ByTier = 1
Rank.Name = "Rank"
With ActiveSheet.DrawingObjects("Rank")
.Placement = xlFreeFloating
.PrintObject = True
End With
ActiveSheet.ChartObjects("Rank").Activate
ActiveSheet.ChartObjects("Rank").Name = "Rank"
ActiveChart.ChartType = xlBarStacked
' Shows Top 10 Ranking Scores, by Tier according to Select Case
ActiveChart.SetSourceData Source:=Sheets(Current).Range("D3:AD3,D5:AD14"), _
PlotBy:=xlColumns
SourceStr = "=" & Current & "'!R5C2:R14C2"
ActiveChart.SeriesCollection(1).XValues = SourceStr
ActiveChart.SeriesCollection(2).XValues = SourceStr

```

```

ActiveChart.SeriesCollection(3).XValues = SourceStr
ActiveChart.SeriesCollection(4).XValues = SourceStr
SendToFile - 5
ActiveChart.SeriesCollection(5).XValues = SourceStr
ActiveChart.SeriesCollection(6).XValues = SourceStr
ActiveChart.SeriesCollection(7).XValues = SourceStr
ActiveChart.SeriesCollection(8).XValues = SourceStr
ActiveChart.SeriesCollection(9).XValues = SourceStr
ActiveChart.SeriesCollection(10).XValues = SourceStr
ActiveChart.SeriesCollection(11).XValues = SourceStr
ActiveChart.SeriesCollection(12).XValues = SourceStr
ActiveChart.SeriesCollection(13).XValues = SourceStr
ActiveChart.SeriesCollection(14).XValues = SourceStr
ActiveChart.SeriesCollection(15).XValues = SourceStr
ActiveChart.SeriesCollection(16).XValues = SourceStr
ActiveChart.SeriesCollection(17).XValues = SourceStr
ActiveChart.SeriesCollection(18).XValues = SourceStr
ActiveChart.SeriesCollection(19).XValues = SourceStr
ActiveChart.SeriesCollection(20).XValues = SourceStr
ActiveChart.SeriesCollection(21).XValues = SourceStr
ActiveChart.SeriesCollection(22).XValues = SourceStr
ActiveChart.SeriesCollection(23).XValues = SourceStr
ActiveChart.SeriesCollection(24).XValues = SourceStr
ActiveChart.SeriesCollection(25).XValues = SourceStr
ActiveChart.SeriesCollection(26).XValues = SourceStr
ActiveChart.SeriesCollection(27).XValues = SourceStr
With ActiveChart
    .HasAxis(xlCategory, xlPrimary) = True
    .HasAxis(xlValue, xlPrimary) = False
    .HasTitle = True
    .ChartTitle.Characters.Text = "Top 10 Airfield Alternatives by All Measures"
End With
' Chart and Plot area formatting
ActiveChart.Location Where:=xlLocationAsObject, Name:=Current
ActiveChart.Axes(xlCategory, xlPrimary).CategoryType = xlAutomatic
With ActiveChart.Axes(xlCategory)
    .HasMajorGridlines = False
    .HasMinorGridlines = False
End With
With ActiveChart.Axes(xlValue)
    .HasMajorGridlines = False
    .HasMinorGridlines = False
End With
ActiveChart.HasLegend = True
ActiveChart.Legend.Select
Selection.Position = xlBottom
ActiveChart.PlotArea.Select
With Selection.Border
    .LineStyle = xlNone
End With
With Selection.Interior
    .ColorIndex = 2
    .PatternColorIndex = 1
    .Pattern = xlSolid
End With
ActiveChart.Axes(xlCategory).Select
With Selection.Border
    .LineStyle = xlNone
End With
With Selection
    .MajorTickMark = xlOutside
    .MinorTickMark = xlNone
    .TickLabelPosition = xlNextToAxis
End With

```

```

With ActiveChart.Axes(xlCategory)
    .CrossesAt = 1
    .TickLabelSpacing = 1
    .TickMarkSpacing = 1
    .AxisBetweenCategories = True
    .ReversePlotOrder = True
End With
ActiveChart.ChartArea.Select
With Selection.Border
    .Weight = 2
    .LineStyle = 1
SendToFile = 6
End With
Selection.Interior.ColorIndex = xlAutomatic
ActiveChart.PlotArea.Select
' Spacing between bars
With ActiveChart.ChartGroups(1)
    .Overlap = 100
    .GapWidth = 50
    .HasSeriesLines = False
End With
'Resize formatting for the All Measures Chart
If ByTier = 1 Then
    ActiveChart.Legend.Select
    Selection.Left = 6
    Selection.Width = 433
    Selection.Height = 88
    Selection.Top = 288
    ActiveChart.PlotArea.Select
    Selection.Width = 426
    Selection.Top = 31
    Selection.Height = 260
End If
Range("A1").Select
ActiveSheet.Visible = True
-----END RANKINGS CHART-----
-----
' Go back to GERBIL
Program.Activate
Application.ScreenUpdating = True
End Sub
Sub WriteScenarioSensitivityToFile()
    ' Writes the Scenario Sensitivity results to a file by copying the sheet and
    changing formatting
    MsgBox "Sheet will be saved to GERBIL MODEL RESULTS.XLS in MyDocuments",
    vboOKOnly, "Saving Sensiti
    vity Results"
    Application.ScreenUpdating = False
    Set Program = ActiveWorkbook
    If Not FileExists("GERBIL Model Results.xls") Then
        If Not WorkbookOpen("GERBIL Model Results.xls") Then
            Set NewBook = Workbooks.Add
            With NewBook
                .Title = "GERBIL Model Results"
                .SaveAs Filename:="GERBIL Model Results.xls"
            End With
        End If
    Else
        If Not WorkbookOpen("GERBIL Model Results.xls") Then
            Workbooks.Open "GERBIL Model Results.xls"
        Else
            Workbooks("GERBIL Model Results.xls").Activate
        End If
    End If
End Sub

```

```

'Copy the sheet to Results File
Program.Sheets("Scenario Sensitivity").Copy After:=Workbooks("GERBIL Model
Results.xls").Sheets(Sheets.Count)
'Delete unnecessary info
ActiveSheet.Unprotect
ActiveSheet.Shapes("Button 1").Select
Selection.Delete
ActiveSheet.Shapes("Picture 2").Select
Selection.Delete
ActiveSheet.Shapes("Text Box 3").Select
Selection.Delete
ActiveWindow.DisplayHeadings = True
Cells.Select
Selection.Interior.ColorIndex = xlNone
SendToFile - 7
Selection.Font.ColorIndex = 0
Range("O1").Select
' Go back to GERBIL
Program.Activate
Application.ScreenUpdating = True
End Sub

Sensitivity - 1
' MODULE FOR SENSITIVITY ANALYSIS
' Much of the Sensitivity Analysis code can be found in the RunModelMenu
Userform code
Public Const SR_Row As Byte = 131 'Reference Row in Weights sheet where
sensitivity results a
re written
'The results are actually written one row below this r
eference
Public Const DMCOL As Byte = 7 ' First column in Airfields Data sheet that
contains the me
asures
Public Const VMCol As Byte = 4 'First column in Values sheet that contains
the measures
Public Const WCol As Byte = 2 'First column for ordered measures in the
Weights sheet
Sub ComputeSensitivityValues(RefRow As Byte, SensRun As Byte)
' RefRow refers to the Row where the global weights to be used in the
calculations are stored in the W
eights sheet
'SensRun is 0 for the run where the measure of interest is 0, SensRun is 1
when the measure of intere
st is 1
Set ValueSheet = Worksheets("Values")
Set DataSheet = Worksheets("Airfields Data")
Set WeightSheet = Worksheets("Weights")
ValueSheet.Activate
'Count the number of alternatives
Count = Application.WorksheetFunction.CountA(DataSheet.Range("A2:A29999"))
' Clear previous values
RowRange = "5:" & Application.WorksheetFunction.Text(Count + 4, 0)
ValueSheet.Rows(RowRange).Delete
i = 1
While DataSheet.Cells(i + 1, 6) = True 'For each Airfield Alternative in Lens
'Alternate code for screening alternatives for sensitivity analysis
' While DataSheet.Cells(i+1,1) <> "" 'Alternately, run through all
Alternatives
'Screen = DataSheet.Cells(i+1,6) = True 'Not in Lens
' OR DataSheet.Cells(i+1, 13) = "None" 'No Dip Clearance
' OR DataSheet.Cells(i+1, 15) = "Unacceptable" ' Dept of State
'If Not(Screen) Then ' In the future build ability to screen for sensit

```

```

ivity analysis
j = i + 4 'j index adjusts the Airfields Data index for the Value sheet
ValueSheet.Cells(j, 2) = DataSheet.Cells(i + 1, 1)
'Calculate all Categorical Values
'Calls the Get_Cat_Values from the Value Calculations Module
With ValueSheet
.Cells(j, VMCol + 6) =
Get_Cat_Values("Diplomatic Clearance", DataSheet.Cells(i + 1, DMCol + 6))
.Cells(j, VMCol + 7) =
Get_Cat_Values("Force Protection", DataSheet.Cells(i + 1, DMCol + 7))
.Cells(j, VMCol + 8) =
Get_Cat_Values("Dept of State", DataSheet.Cells(i + 1, DMCol + 8))
.Cells(j, VMCol + 9) =
Get_Cat_Values("Military Cooperation", DataSheet.Cells(i + 1, DMCol + 9))
' Break for 4 continuous value functions
.Cells(j, VMCol + 14) =
Get_Cat_Values("Lodging", DataSheet.Cells(i + 1, DMCol + 14))
.Cells(j, VMCol + 15) =
Get_Cat_Values("Dining", DataSheet.Cells(i + 1, DMCol + 15))
.Cells(j, VMCol + 16) =
Get_Cat_Values("Medical", DataSheet.Cells(i + 1, DMCol + 16))
.Cells(j, VMCol + 17) =
Get_Cat_Values("Communications", DataSheet.Cells(i + 1, DMCol + 17))
.Cells(j, VMCol + 18) =
Get_Cat_Values("Power", DataSheet.Cells(i + 1, DMCol + 18))
.Cells(j, VMCol + 19) =
Get_Cat_Values("Potable Water", DataSheet.Cells(i + 1, DMCol + 19))
.Cells(j, VMCol + 20) =
Get_Cat_Values("Sewer", DataSheet.Cells(i + 1, DMCol + 20))
.Cells(j, VMCol + 21) =
Get_Cat_Values("Mountainous", DataSheet.Cells(i + 1, DMCol + 21))
Sensitivity - 2
.Cells(j, VMCol + 22) =
Get_Cat_Values("Altitude", DataSheet.Cells(i + 1, DMCol + 22))
.Cells(j, VMCol + 23) =
Get_Cat_Values("Weather", DataSheet.Cells(i + 1, DMCol + 23))
.Cells(j, VMCol + 24) =
Get_Cat_Values("Temperature", DataSheet.Cells(i + 1, DMCol + 24))
.Cells(j, VMCol + 25) =
Get_Cat_Values("Urban Areas", DataSheet.Cells(i + 1, DMCol + 25))
.Cells(j, VMCol + 26) =
Get_Cat_Values("Terrain", DataSheet.Cells(i + 1, DMCol + 26))
End With
'Calculate Continuous Measure Values using Value Function Sheet
'Calls the Get_Cont_Values from the Value Calculations Module
With ValueSheet
.Cells(j, VMCol) =
Get_Cont_Values("MOG", DataSheet.Cells(i + 1, DMCol))
.Cells(j, VMCol + 1) =
Get_Cont_Values("Critical Leg", DataSheet.Cells(i + 1, DMCol + 1))
.Cells(j, VMCol + 2) =
Get_Cont_Values("Delta Flight Length", DataSheet.Cells(i + 1, DMCol + 2))
.Cells(j, VMCol + 3) =
Get_Cont_Values("Alternate Airfields", DataSheet.Cells(i + 1, DMCol + 3))
.Cells(j, VMCol + 4) =
Get_Cont_Values("Fuel Storage", DataSheet.Cells(i + 1, DMCol + 4))
.Cells(j, VMCol + 5) =
Get_Cont_Values("Fuel Resupply", DataSheet.Cells(i + 1, DMCol + 5))
' Break in Continuous VFs
.Cells(j, VMCol + 10) =
Get_Cont_Values("Seaport", DataSheet.Cells(i + 1, DMCol + 10))
.Cells(j, VMCol + 11) =
Get_Cont_Values("Railroad", DataSheet.Cells(i + 1, DMCol + 11))
.Cells(j, VMCol + 12) =

```

```

Get_Cont_Values("Road System", DataSheet.Cells(i + 1, DMCol + 12))
.Cells(j, VMCol + 13) =
Get_Cont_Values("Commercial Airport", DataSheet.Cells(i + 1, DMCol + 13))
End With
'WEIGHT THE VALUES!
'Multiply Measures by Global Weights stored on Weights Sheet B106:AB106
For wt = 0 To 26 'for all 27 measures
Value = ValueSheet.Cells(j, VMCol + wt).Value
Weight = WeightSheet.Cells(RefRow, WCol + wt).Value
WeightedVal = Value * Weight
ValueSheet.Cells(j, VMCol + wt) = WeightedVal 'write results to the value
sheet
Next wt
' Copy formulas from the BEST POSSIBLE SCORE row to get the total value score
TOT = "C4:C" & Application.WorksheetFunction.Text(j, 0)
Range("C4").Copy Range(TOT)
' Write the results to the Weight Sheet for Sensitivy Analysis
WeightSheet.Cells(SR_Row + i, 1) = ValueSheet.Cells(i + 4, 2) 'Write airfield
name
WeightSheet.Cells(SR_Row + i, SensRun + 2) = ValueSheet.Cells(i + 4, 3) 'Write
Value Scores
i = i + 1 'look at next Alternative
Wend 'Until alternative not in lens (or alternately, no more alternatives if
using
screen)
'End if 'If Not Screened
Worksheets("Values").Activate
Columns("D:BC").AutoFit
'Setting the number format for all the value scores
Count = Application.WorksheetFunction.CountA(ValueSheet.Range("B5:B65536"))
SortRange = "BB" & Application.WorksheetFunction.Text(Count + 4, 0)
Range("B5:" & SortRange).Select
Selection.NumberFormat = "0.0000"
End Sub

Uncertainty - 1
Public Const UCol As Byte = 3 'First Col where measures begin in Uncertainty
Data
Public Const TempRow As Byte = 3 'Row where current data input is kept in
Uncertainty Data
' UNCERTAINTY Module to compute the expected value score if a measure given an
uncertain distribution
' A lot of code exists in the AirfieldInput, EmpiricalDist, and TriangleDist
Userforms
' A TempRow in the Uncertainty Data sheet is used to initialize and store
uncertainty distributions fo
r the
' airfield that is currently being added or edited in the AirfieldInput form
' Although all the measures have an uncertainty distribution associated with
it
' They are not active until the Assign Probability checkbox in the
AirfieldInput userform is marked
Sub InitializeTempRow()
' Initializes row 3 as the Temp Row for uncertainty info
' Continuous measures are initialized with the hi as the max, the lo as the
min, and (hi+lo)/2 as the
mode
' Categorical measures are initialized with each possible category equally
likely
Set USheet = Worksheets("Uncertainty Data")
USheet.Unprotect
USheet.Range("B3:GN3").Clear 'Clear the previous Temp values
i = UCol
While USheet.Cells(1, i) <> "" 'For all measures

```

```

Measure = USheet.Cells(1, i)
If IsContVF(Measure) Then 'If at a Continuous measure
Call GetContVFRange(Measure, lo, hi)
USheet.Cells(TempRow, i) = lo
USheet.Cells(TempRow, i + 1) = (lo + hi) / 2
USheet.Cells(TempRow, i + 2) = hi
i = i + 3 'Step to next measure
Else: 'If at a Categorical measure
Count = 0
For j = 0 To 9 'Determine the number of categories for that measure
Count = Count + 1
End If
Next j
Prob = 1 / Count 'Prepare an equally likely probability
For j = 0 To Count - 1 'Set all possible categories to equally likely
USheet.Cells(TempRow, i + j) = Prob
Next j
i = i + 10 'Step to next measure
End If
Wend
USheet.Columns("A:GN").AutoFit
USheet.Columns("A:GN").HorizontalAlignment = xlCenter
USheet.Protect
End Sub
Function ExpectedValue(Measure, Index)
'Determines the expected value for each measure
' The EV for Continous VF is the EV of a Triangle Dist (MinVal + ModeVal +
MaxVal)/3
' The EV for Categorical VF is Sum (Val+i * Prob_i) for i = 1 to # categories
(max 10)
' Used by the ComputeMeasureValue subroutine in the ValueCalculations module
Set USheet = Worksheets("Uncertainty Data")
i = UCol 'Find Measure in Uncertainty Data
While USheet.Cells(1, i) <> Measure
i = i + 1
Wend
If IsContVF(Measure) Then
MinVal = Get_Cont_Values(Measure, USheet.Cells(Index, i))
ModeVal = Get_Cont_Values(Measure, USheet.Cells(Index, i + 1))
MaxVal = Get_Cont_Values(Measure, USheet.Cells(Index, i + 2))
ExpectedValue = (MinVal + ModeVal + MaxVal) / 3
Else 'Categorical Value Function
EV = 0
For j = 0 To 9
EV = EV + Get_Cat_Values(Measure, USheet.Cells(2, i + j)) *
USheet.Cells(Index, i + j)
Next j
Uncertainty = 2
ExpectedValue = EV
End If
End Function

ValueCalculations - 1
Public Const VMCOL As Byte = 4 ' First Column of measures on Value Sheet
Public Const DMCOL As Byte = 7 ' First Column of measures on Data Sheet
Public Const NumToRank As Byte = 10 'NOT USED, the number of alternatives to
show in the Rankings
chart
Function Get_Cat_Values(MeasureSheetName, CatVal)
' Finds the value of a categorical measure by searching the value function
sheet
' Called by ComputeValueMeasure in this Module

```

```

' Also called by the ExpectedValue function in the Uncertainty Module
' Also called by ComputeSensitivityValues in the Sensitivity Module
Get_Cat_Values = 0
For i = 0 To 9
CatCompare = Worksheets(MeasureSheetName).Cells(i + 3, 12) 'The alternatives
data point f
or this measure
If CatVal = CatCompare Then
Get_Cat_Values = Worksheets(MeasureSheetName).Cells(i + 3, 13) 'The value
score for that
category
End If
Next i
End Function
Function Get_Cont_Values(MeasureSheetName, MeasureData)
' Finds the value of a continuous measure by entering the data value in the
value function sheet for e
valuation
' Called by ComputeValueMeasure in this Module
' Also called by the ExpectedValue function in the Uncertainty Module
' Also called by ComputeSensitivityValues in the Sensitivity Module
Worksheets(MeasureSheetName).Range("M23") = MeasureData
Get_Cont_Values = Worksheets(MeasureSheetName).Range("M24")
End Function
Sub ComputeValueScores()
' Used to compute all the value scores and post results in the Values
worksheet
' Called by the CreateRankings subroutine in this Module
Set VS = Worksheets("Values")
Set DataSheet = Worksheets("Airfields Data")
Set WeightSheet = Worksheets("Weights")
Set ODSheet = Worksheets("Orig-Dest Airfields")
Set Current = ActiveSheet
VSActivate
VS.Unprotect
'Count the number of alternatives
Count = Application.WorksheetFunction.CountA(DataSheet.Range("A2:A29999"))
'Clear previous values
VS.Rows("5:65356").Delete
VS.Range("E1") = ""
VS.Range("L1") = ""
For i = 1 To Count 'For each Airfield Alternative
j = i + 4 'sync the index for the Value Sheet
VS.Cells(j, 2) = DataSheet.Cells(i + 1, 1)
'Mark airfields with potential screening criteria
If Not (DataSheet.Cells(i + 1, 6)) Then ' If Alternative is not In the Lens,
mark with ast
erisk
VS.Cells(j, 2) = "*" & VS.Cells(j, 2)
End If
If (DataSheet.Cells(i + 1, DMCOL + 6) = "None") Then 'If Alternative Dip
Clearance is NONE
VS.Cells(j, 2) = "#" & VS.Cells(j, 2)
End If
If (DataSheet.Cells(i + 1, DMCOL + 8) = "Unacceptable") Then 'If Alternative
Dept of State is
UNACCEPTABLE
VS.Cells(j, 2) = "!" & VS.Cells(j, 2)
End If
ValueCalculations - 2
' Call GetContVFRange("Critical Leg", lo, hi) ' Get the minimum critical leg
' If (DataSheet.Cells(i + 1, DMCOL + 1) < lo) Then 'If less than minimum
critical leg, enrou
te not needed

```

```

' VS.Cells(j, 2) = VS.Cells(j, 2) & "@"
' End If
'Calculate Measure Values using Value Function Sheet
'Ensure that the CORRECT MeasureName used on the VF Sheet is used
Call ComputeMeasureValue(i, j, "MOG", 0)
Call ComputeMeasureValue(i, j, "Critical Leg", 1)
Call ComputeMeasureValue(i, j, "Delta Flight Length", 2)
Call ComputeMeasureValue(i, j, "Alternate Airfields", 3)
Call ComputeMeasureValue(i, j, "Fuel Storage", 4)
Call ComputeMeasureValue(i, j, "Fuel Resupply", 5)
Call ComputeMeasureValue(i, j, "Diplomatic Clearance", 6)
Call ComputeMeasureValue(i, j, "Force Protection", 7)
Call ComputeMeasureValue(i, j, "Dept of State", 8)
Call ComputeMeasureValue(i, j, "Military Cooperation", 9)
Call ComputeMeasureValue(i, j, "Seaport", 10)
Call ComputeMeasureValue(i, j, "Railroad", 11)
Call ComputeMeasureValue(i, j, "Road System", 12)
Call ComputeMeasureValue(i, j, "Commercial Airport", 13)
Call ComputeMeasureValue(i, j, "Lodging", 14)
Call ComputeMeasureValue(i, j, "Dining", 15)
Call ComputeMeasureValue(i, j, "Medical", 16)
Call ComputeMeasureValue(i, j, "Communications", 17)
Call ComputeMeasureValue(i, j, "Power", 18)
Call ComputeMeasureValue(i, j, "Potable Water", 19)
Call ComputeMeasureValue(i, j, "Sewer", 20)
Call ComputeMeasureValue(i, j, "Mountainous", 21)
Call ComputeMeasureValue(i, j, "Altitude", 22)
Call ComputeMeasureValue(i, j, "Weather", 23)
Call ComputeMeasureValue(i, j, "Temperature", 24)
Call ComputeMeasureValue(i, j, "Urban Areas", 25)
Call ComputeMeasureValue(i, j, "Terrain", 26)
'WEIGHT THE VALUE SCORES
'Multiply Measures by Global Weights stored on Weights Sheet B106:AB106
WRow = 106
For wt = 0 To 26
Value = VS.Cells(j, VMCol + wt).Value
Weight = WeightSheet.Cells(WRow, WCol + wt).Value
WeightedVal = Value * Weight
VS.Cells(j, VMCol + wt) = WeightedVal
Next wt
' Copy formulas from Best Possible Score row to each alternative
' This computes the total value score and value scores by tier
Row = "AE" & Application.WorksheetFunction.Text(j, 0)
Worksheets("Values").Activate
Range("$AE$4:$BB$4").Copy Range(Row)
TOT = "C4:C" & Application.WorksheetFunction.Text(j, 0)
Range("C4").Copy Range(TOT)
Next i
Worksheets("Values").Activate
Columns("D:BC").AutoFit
'Set the number format for all value scores
Count = Application.WorksheetFunction.CountA(VS.Range("B5:B65536"))
SortRange = "BB" & Application.WorksheetFunction.Text(Count + 4, 0)
Range("B5:" & SortRange).Select
Selection.NumberFormat = "0.0000"
With Selection
.Font.Size = 10
.Interior.ColorIndex = xlNone
End With
Selection.FormatConditions.Delete
Selection.FormatConditions.Add Type:=xlCellValue, Operator:=xlEqual, _
Formula1:="=B$4"
Selection.FormatConditions(1).Font.ColorIndex = 14
Selection.FormatConditions.Add Type:=xlCellValue, Operator:=xlEqual, _

```

```

ValueCalculations = 3
Formula1:="0"
Selection.FormatConditions(2).Font.ColorIndex = 3
'Sort all the value scores according to the Total Score
Selection.Sort Key1:=Range("C5"), Order1:=xlDescending, Header:=xlNo, _
OrderCustom:=1, MatchCase:=False, Orientation:=xlTopToBottom, _
DataOption1:=xlSortNormal
'Write the rank number for the ranked alternatives and add the total score to
the alternative
name
'This aids when displaying the chart
For i = 1 To Count
Cells(i + 4, 1) = i
Cells(i + 4, 2) = Cells(i + 4, 2) & " " & Application.WorksheetFunction.Text _
(Cells(i + 4, 3), "0.000")
Next i
Columns("A:BB").AutoFit
Range("A1").Select
OriginIndex = ODSheet.Cells(1, 2)
DestIndex = ODSheet.Cells(2, 2)
VS.Range("E1") = ODSheet.Cells(FirstOD + OriginIndex - 1, 1)
VS.Range("L1") = ODSheet.Cells(FirstOD + DestIndex - 1, 11)
VS.Protect
CurrentActivate
End Sub
Sub CreateRankings(ByTier) ' CREATES THE CHART IN "Rankings" WORKSHEET
' By Tier is set by the option selected in the RunModelMenu Userform or by the
Tier spinner on the Ran
kings sheet
Set ValueSheet = Worksheets("Values")
Set DataSheet = Worksheets("Airfields Data")
Set WeightSheet = Worksheets("Weights")
Set RankSheet = Worksheets("Rankings")
Set ODSheet = Worksheets("Orig-Dest Airfields")
ValueSheet.Unprotect
RankSheet.Unprotect
Call ComputeValueScores
Count = Application.WorksheetFunction.CountA(ValueSheet.Range("B4:B65536"))
If Count = 0 Then
MsgBox "There are no alternatives to rank", vbOKOnly, "No Alternatives"
Else:
OriginIndex = ODSheet.Cells(1, 2)
DestIndex = ODSheet.Cells(2, 2)
' -----CREATE RANKINGS CHART-----
-----
RankSheet.Activate
RankSheet.Range("L12") = ODSheet.Cells(FirstOD + OriginIndex - 1, 1)
RankSheet.Range("L15") = ODSheet.Cells(FirstOD + DestIndex - 1, 11)
ActiveSheet.ChartObjects.Delete 'Delete Previous Sheet
Set Rank = RankSheet.ChartObjects.Add(5, 5, 475, 385)
Rank.Name = "Rank"
With RankSheet.DrawingObjects("Rank")
.Placement = xlFreeFloating
.PrintObject = True
End With
RankSheet.ChartObjects("Rank").Activate
RankSheet.ChartObjects("Rank").Name = "Rank"
ActiveChart.ChartType = xlBarStacked
' Shows Top 10 Ranking Scores, by Tier according to Select Case
RankSheet.Range("c3") = ByTier
ValueCalculations = 4
Select Case ByTier
' Case 1-----
Case 1 'All Measures

```

```

ActiveChart.SetSourceData Source:=Sheets("Values").Range("D3:AD3,D5:AD14"), _
PlotBy:=xlColumns
ActiveChart.SeriesCollection(1).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(2).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(3).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(4).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(5).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(6).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(7).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(8).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(9).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(10).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(11).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(12).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(13).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(14).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(15).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(16).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(17).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(18).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(19).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(20).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(21).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(22).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(23).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(24).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(25).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(26).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(27).XValues = "=Values!R5C2:R14C2"
With ActiveChart
    .HasAxis(xlCategory, xlPrimary) = True
    .HasAxis(xlValue, xlPrimary) = False
    .HasTitle = True
    .ChartTitle.Characters.Text = "Top 10 Airfield Alternatives by All Measures"
End With
' Case 2-----
Case 2 'By Tier 3
ActiveChart.SetSourceData Source:=Sheets("Values").Range("AE3:AT3,AE5:AT14"), _
PlotBy:=xlColumns
ActiveChart.SeriesCollection(1).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(2).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(3).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(4).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(5).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(6).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(7).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(8).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(9).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(10).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(11).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(12).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(13).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(14).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(15).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(16).XValues = "=Values!R5C2:R14C2"
With ActiveChart
    .HasAxis(xlCategory, xlPrimary) = True
    .HasAxis(xlValue, xlPrimary) = False
    .HasTitle = True
    .ChartTitle.Characters.Text = "Top 10 Airfield Alternatives by Tier 3"
End With
' Case 3-----
Case 3 'By Tier 2

```

```

ActiveChart.SetSourceData Source:=Sheets("Values").Range("AU3:AZ3,AU5:AZ14"),
PlotBy:=xlColumns
ActiveChart.SeriesCollection(1).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(2).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(3).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(4).XValues = "=Values!R5C2:R14C2"
ValueCalculations = 5
ActiveChart.SeriesCollection(5).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(6).XValues = "=Values!R5C2:R14C2"
With ActiveChart
    .HasAxis(xlCategory, xlPrimary) = True
    .HasAxis(xlValue, xlPrimary) = False
    .HasTitle = True
    .ChartTitle.Characters.Text = "Top 10 Airfield Alternatives by Tier 2"
End With
' Case 4-----
Case 4 ' By Tier 1
ActiveChart.SetSourceData Source:=Sheets("Values").Range("BA3:BB3,BA5:BB14"),
PlotBy:=xlColumns
ActiveChart.SeriesCollection(1).XValues = "=Values!R5C2:R14C2"
ActiveChart.SeriesCollection(2).XValues = "=Values!R5C2:R14C2"
With ActiveChart
    .HasAxis(xlCategory, xlPrimary) = True
    .HasAxis(xlValue, xlPrimary) = False
    .HasTitle = True
    .ChartTitle.Characters.Text = "Top 10 Airfield Alternatives by Tier 1"
End With
End Select
' Chart and Plot area formatting
ActiveChart.Location Where:=xlLocationAsObject, Name:="Rankings"
ActiveChart.Axes(xlCategory, xlPrimary).CategoryType = xlAutomatic
With ActiveChart.Axes(xlCategory)
    .HasMajorGridlines = False
    .HasMinorGridlines = False
End With
With ActiveChart.Axes(xlValue)
    .HasMajorGridlines = False
    .HasMinorGridlines = False
End With
ActiveChart.HasLegend = True
ActiveChart.Legend.Select
Selection.Position = xlBottom
ActiveChart.PlotArea.Select
With Selection.Border
    .LineStyle = xlNone
End With
With Selection.Interior
    .ColorIndex = 2
    .PatternColorIndex = 1
    .Pattern = xlSolid
End With
ActiveChart.Axes(xlCategory).Select
With Selection.Border
    .LineStyle = xlNone
End With
With Selection
    .MajorTickMark = xlOutside
    .MinorTickMark = xlNone
    .TickLabelPosition = xlNextToAxis
End With
With ActiveChart.Axes(xlCategory)
    .CrossesAt = 1

```

```

.TickLabelSpacing = 1
.TickMarkSpacing = 1
.AxisBetweenCategories = True
.ReversePlotOrder = True
.TickLabels.Font.Size = 8
End With
ActiveChart.ChartArea.Select
With Selection.Border
.Weight = 2
.LineStyle = 0
End With
Selection.Interior.ColorIndex = xlAutomatic
ActiveChart.PlotArea.Select
' Spacing between bars
ValueCalculations = 6
With ActiveChart.ChartGroups(1)
.Overlap = 100
.GapWidth = 50
.HasSeriesLines = False
End With
'Resize formatting for the All Measures Chart
If ByTier = 1 Then
ActiveChart.Legend.Select
Selection.Left = 6
Selection.Width = 433
Selection.Height = 88
Selection.Top = 288
ActiveChart.PlotArea.Select
Selection.Width = 426
Selection.Top = 31
Selection.Height = 260
End If
-----END RANKINGS CHART-----
-----
RankSheet.Visible = True
End If ' If there are any alternatives
Range("O1").Select
ValueSheet.Protect
RankSheet.Protect
End Sub
Sub TierSpinner()
' Allows user to view rankings by higher or lower tiers
' The Tier control is tied to cell C3
' Right click on the spinner in Rankings sheet to see the control formatting.
Min Value is 1, Max Val
ue is 4
Application.ScreenUpdating = False
ByTier = Range("c3")
Range("c3").Select
Call CreateRankings(ByTier)
Application.ScreenUpdating = True
End Sub
Sub ComputeMeasureValue(Alternative, ValueSheetIndex, MeasureName,
MeasureIndex)
' Called by ComputeValueScores in this Module
' Uses ExpectedValue function from Uncertainty Module
Set DataSheet = Worksheets("Airfields Data")
Set VS = Worksheets("Values")
' Application.ScreenUpdating = False
' If an uncertainty distribution is assigned to the measure
If DataSheet.Cells(Alternative + 1, DMCol + MeasureIndex) = "?" Then
VS.Cells(ValueSheetIndex, VMCol + MeasureIndex) = ExpectedValue(MeasureName,
Alternative +
3)

```

```

Else 'No uncertainty, determine the value from function
If IsContVF(MeasureName) Then 'Continuous Measure
If (MeasureName = "Critical Leg") Then
Call GetContVFRange("Critical Leg", lo, hi)
If DataSheet.Cells(Alternative + 1, DMCOL + MeasureIndex) < lo Then
VS.Cells(ValueSheetIndex, VMCOL + MeasureIndex) = 0
Else
VS.Cells(ValueSheetIndex, VMCOL + MeasureIndex) =
Get_Cont_Values(MeasureName, DataSheet.Cells(Alternative + 1, DMCOL + Meas
ureIndex))
End If
Else 'Continuous VF but not Critical Leg
VS.Cells(ValueSheetIndex, VMCOL + MeasureIndex) =
Get_Cat_Values(MeasureName, DataSheet.Cells(Alternative + 1, DMCOL + Meas
ureIndex))
End If
Else 'Categorical Measure
VS.Cells(ValueSheetIndex, VMCOL + MeasureIndex) =
Get_Cat_Values(MeasureName, DataSheet.Cells(Alternative + 1, DMCOL +
MeasureIndex))
ValueCalculations = 7
End If
End If
' Application.ScreenUpdating = True
End Sub

ValueFunctions = 1
' VALUE FUNCTION CHART SUBROUTINES
' Much of this code was adapted from Lt Col Jeffrey Weir, AFIT OR Department,
HierarchyBuilder.XLA
Function IsContVF(Measure) As Boolean
' Returns TRUE if Measure is Continuous, False otherwise, assumed Categorical
M = Measure
IsContVF = False
If M = "MOG" Or
M = "Alternate Airfields" Or _
M = "Fuel Storage" Or _
M = "Fuel Resupply" Or _
M = "Critical Leg" Or
M = "Delta Flight Length" Or _
M = "Seaport" Or _
M = "Railroad" Or _
M = "Road System" Or _
M = "Commercial Airport" Then
IsContVF = True
End If
End Function
Sub Adjust_Graph_new_bounds()
'
Application.ScreenUpdating = False
ActiveSheet.Unprotect
sheetname = ActiveSheet.Name
oldlowerbound = Sheets(sheetname).Cells(100, 1)
oldupperbound = Sheets(sheetname).Cells(110, 15)
ActiveSheet.ChartObjects("Chart1").Activate
ActiveChart.Axes(xlCategory).Select
With ActiveChart.Axes(xlCategory)
.MinimumScale = Sheets(sheetname).Cells(14, 13)
.MaximumScale = Sheets(sheetname).Cells(15, 13)
.MinorUnitIsAuto = True
.MajorUnitIsAuto = True
.Crosses = xlAutomatic
.ReversePlotOrder = False
.ScaleType = xlLinear

```

```

.DisplayUnit = xlNone
End With
ActiveWindow.Visible = False
Range("L14").Select
For j = 1 To 15 Step 2
If (ActiveSheet.Cells(110, j) = oldupperbound) Then ActiveSheet.Cells(110, j)
= Sheets(sheetna
me).Cells(15, 13)
If (ActiveSheet.Cells(100, j) = oldlowerbound) Then ActiveSheet.Cells(100, j)
= Sheets(sheetna
me).Cells(14, 13)
Next j
Call ProtectOneMeasure
Application.ScreenUpdating = True
End Sub
Sub Adjust_Reference_Pt_inc()
Score = ActiveSheet.Cells(3, 13)
Value = ActiveSheet.Cells(4, 13)
'find segemngt reference point is located on
ActiveSheet.Unprotect
For j = 1 To 15 Step 2
If (Score <= ActiveSheet.Cells(110, j) And Score >= ActiveSheet.Cells(100, j))
Then Number = j
Next j
rho = ActiveSheet.Cells(99, Number)
high = ActiveSheet.Cells(110, Number)
low = ActiveSheet.Cells(100, Number)
If (Abs(rho) > 10000 * high) Then
ActiveSheet.Cells(99, Number) = 150 * high
rho = ActiveSheet.Cells(99, Number)
End If
If ((Score <= high) And (Score >= low)) Then
If (Value > ((Score - low) / (high - low)) * (ActiveSheet.Cells((Number + 1) /
2 + 120, 1) - Activ
ValueFunctions - 2
eSheet.Cells((Number + 1) / 2 + 119, 1)) + (ActiveSheet.Cells((Number + 1) / 2
+ 119, 1))) Then
If (ActiveSheet.Cells(99, Number) < 0) Then ActiveSheet.Cells(99, Number) =
ActiveSheet.Cells(
99, Number) * -1
Range(Chr(64 + Number + 1) & "99").GoalSeek Goal:=Value,
ChangingCell:=Range(Chr(64 + Number)
& "99")
Else
If (ActiveSheet.Cells(99, Number) > 0) Then ActiveSheet.Cells(99, Number) =
ActiveSheet.Cells(
99, Number) * -1
Range(Chr(64 + Number + 1) & "99").GoalSeek Goal:=Value,
ChangingCell:=Range(Chr(64 + Number)
& "99")
End If
Else
If ((Score > high) Or (Score < low)) Then MsgBox "Score out of bounds",
vbOKOnly, "Invalid Input"
If ((Value > 1) Or (Value < 0)) Then MsgBox "Value out of bounds", vbOKOnly,
"Invalid Input"
End If
Call ProtectOneMeasure
End Sub
Sub Adjust_Reference_Pt_dec()
Score = ActiveSheet.Cells(3, 13)
Value = ActiveSheet.Cells(4, 13)
ActiveSheet.Unprotect
'find segemngt reference point is located on

```

```

For j = 1 To 15 Step 2
If (Score <= ActiveSheet.Cells(110, j) And Score >= ActiveSheet.Cells(100, j))
Then Number = j
Next j
rho = ActiveSheet.Cells(99, Number)
high = ActiveSheet.Cells(110, Number)
low = ActiveSheet.Cells(100, Number)
If (Abs(rho) > 10000 * high) Then
ActiveSheet.Cells(99, Number) = 150 * high
rho = ActiveSheet.Cells(99, Number)
End If
If ((Score <= high) And (Score >= low)) Then
If (Value < ((high - Score) / (high - low)) * (ActiveSheet.Cells((Number + 1) / 2 + 120, 1) - ActiveSheet.Cells((Number + 1) / 2 + 119, 1)) + (ActiveSheet.Cells((Number + 1) / 2 + 119, 1))) Then
If (ActiveSheet.Cells(99, Number) > 0) Then ActiveSheet.Cells(99, Number) =
ActiveSheet.Cells(99, Number) * -1
Range(Chr(64 + Number + 1) & "99").GoalSeek Goal:=Value,
ChangingCell:=Range(Chr(64 + Number) & "99")
Else
If (ActiveSheet.Cells(99, Number) < 0) Then ActiveSheet.Cells(99, Number) =
ActiveSheet.Cells(99, Number) * -1
Range(Chr(64 + Number + 1) & "99").GoalSeek Goal:=Value,
ChangingCell:=Range(Chr(64 + Number) & "99")
End If
Else
If ((Score > high) Or (Score < low)) Then MsgBox "Score out of bounds",
vbOKOnly, "Invalid Input"
If ((Value > 1) Or (Value < 0)) Then MsgBox "Value out of bounds", vbOKOnly,
"Invalid Input"
End If
Call ProtectOneMeasure
End Sub
Sub ClickOnAddOrDelete()
Msg = "Adding/Deleting Categories may affect current airfield alternatives" &
vbCrLf & vbCrLf
& "If deleting a category, alternatives with this value will receive" & vbCrLf
& "a zero score for this measure." & vbCrLf & vbCrLf
& "Are you sure you want to Add or Delete Categories?" &
Ans = MsgBox(Msg, vbYesNo, "Confirm Add or Delete Categories")
If (Ans = vbYes) Then
Call Adjust_graph_category_add_or_delete
End If
End Sub
Sub Adjust_graph_category_add_or_delete()
Application.ScreenUpdating = False
ValueFunctions = 3
ActiveSheet.Unprotect
sheetname = ActiveSheet.Name
Sheets(sheetname).ChartObjects("Chart1").Activate
lastxval = 3
While (Sheets(sheetname).Cells(lastxval, 13) <> "")
lastxval = lastxval + 1
Wend
lastxval = lastxval - 1
ActiveChart.SeriesCollection(1).XValues = "=" & sheetname & "'!R3C12:R" &
lastxval & "C12"

```

```

ActiveChart.SeriesCollection(1).Values = "=" & sheetname & "'!R3C13:R" &
lastxval & "C13"
Sheets(sheetname).Range("E5").Select
'Update categories in Uncertainty Data sheet
CatMeasure = ActiveSheet.Name
For j = 1 To 200
If Worksheets("Uncertainty Data").Cells(1, j) = CatMeasure Then
For k = 1 To 10
Worksheets("Uncertainty Data").Unprotect
Worksheets("Uncertainty Data").Cells(2, j + k - 1) = ActiveSheet.Cells(k + 2,
12)
Worksheets("Uncertainty Data").Protect
Next k
End If
Next j
Call ProtectOneMeasure
Application.ScreenUpdating = True
End Sub
Sub Break_Reference_Pt()
Score = ActiveSheet.Cells(3, 13)
Value = ActiveSheet.Cells(4, 13)
sheetname = ActiveSheet.Name
ActiveSheet.Unprotect
'find segment reference point is located on
For j = 1 To 15 Step 2
If (Score <= ActiveSheet.Cells(110, j) And Score >= ActiveSheet.Cells(100, j)) Then
Number = j
Next j
If (ActiveSheet.Cells((Number + 1) / 2 + 120, 1) = 1) Or
(ActiveSheet.Cells((Number + 1) / 2 + 120, 1)
= 0) Then
ActiveSheet.Cells((Number + 1) / 2 + 120, 1) = Value
ActiveSheet.Cells(110, Number + 2) = ActiveSheet.Cells(110, Number)
ActiveSheet.Cells(110, Number) = Score
Else
Count = 127
While (Count > 120 + (Number + 1) / 2)
ActiveSheet.Cells(Count, 1) = ActiveSheet.Cells(Count - 1, 1)
ActiveSheet.Cells(110, (Count - 119) * 2 - 1) = ActiveSheet.Cells(110, (Count -
120) * 2 - 1)
Count = Count - 1
Wend
ActiveSheet.Cells(110, (Count - 119) * 2 - 1) = ActiveSheet.Cells(110, (Count -
120) * 2 - 1)
ActiveSheet.Cells(120 + (Number + 1) / 2, 1) = Value
ActiveSheet.Cells(110, Number) = Score
End If
Call ProtectOneMeasure
End Sub
Sub DeleteE_Break_Pt()
On Error GoTo mes
Score = ActiveSheet.Cells(3, 13)
Value = ActiveSheet.Cells(4, 13)
Number = 0
ActiveSheet.Unprotect
ValueFunctions = 4
'find if break point exists
For j = 1 To 15 Step 2
If (Score = ActiveSheet.Cells(110, j) And (Value = ActiveSheet.Cells(110, j +
1))) Then Number = j
Next j
mes:
If (Number > 0) Then
Count = 120 + (Number + 1) / 2

```

```

While (Count < 128)
ActiveSheet.Cells(Count, 1) = ActiveSheet.Cells(Count + 1, 1)
ActiveSheet.Cells(110, (Count - 120) * 2 - 1) = ActiveSheet.Cells(110, (Count
- 119) * 2 - 1)
Count = Count + 1
Wend
Else
MsgBox "Reference point is not a Break Point. To find a break points value,
move cursor over the
break point."
End If
Call ProtectOneMeasure
End Sub
Sub ProtectMeasures()
'
' Macro2 Macro
' Macro recorded 2/24/2006 by Alex Miravite
'
'
' Application.ScreenUpdating = False
For i = MeasureSheets To Sheets.Count
Sheets(i).Activate
Call ProtectOneMeasure
Next i
' Application.ScreenUpdating = True
End Sub
Sub ProtectOneMeasure()
'Locks most of the Value Function sheet and sets limited scroll area
Application.ScreenUpdating = False
ActiveSheet.Unprotect
If (Range("L3") = "Score") Then
Range("M3:M4,M14:M15,M23").Select
Selection.Locked = False
Selection.FormulaHidden = False
Range("L3:L4,L14:L15,L23:L24").Select
Selection.Locked = True
Selection.FormulaHidden = True
Range("o1").Select
Else
Range("L3:M12").Select
Selection.Locked = False
Selection.FormulaHidden = False
Range("o1").Select
End If
ActiveSheet.ScrollArea = "A1:S29"
Range("M4").Select
ActiveSheet.Protect
ActiveSheet.EnableSelection = xlUnlockedCells
Application.ScreenUpdating = True
End Sub
Sub PrintALLValueFunctions()
Msg = "This will print all 27 Value functions." & vbCrLf & _
"Is your printer paper ready?" & vbCrLf & vbCrLf & _
"Do you want to continue?"
ValueFunctions - 5
Ans = MsgBox(Msg, vbYesNo, "Print All Value Functions")
If (Ans = vbYes) Then
Application.ScreenUpdating = False
Current = ActiveSheet.Name
For i = MeasureSheets To Sheets.Count ' MeasureSheets defined in Defaults
Module
Sheets(i).Activate
Sheets(i).Visible = True
ActiveSheet.PageSetup.PrintArea = "A1:N30"

```

```

With ActiveSheet.PageSetup
    .LeftHeader = ""
    .CenterHeader = ""
    .RightHeader = ""
    .LeftFooter = ""
    .CenterFooter = ""
    .RightFooter = ""
    .LeftMargin = Application.InchesToPoints(0.75)
    .RightMargin = Application.InchesToPoints(0.75)
    .TopMargin = Application.InchesToPoints(1)
    .BottomMargin = Application.InchesToPoints(1)
    .HeaderMargin = Application.InchesToPoints(0.5)
    .FooterMargin = Application.InchesToPoints(0.5)
    .PrintHeadings = False
    .PrintGridlines = False
    .PrintComments = xlPrintNoComments
    .CenterHorizontally = True
    .CenterVertically = True
    .Orientation = xlLandscape
    .Draft = False
    .PaperSize = xlPaperLetter
    .FirstPageNumber = xlAutomatic
    .Order = xlDownThenOver
    .BlackAndWhite = False
    .Zoom = False
    .FitToPagesWide = 1
    .FitToPagesTall = 1
    .PrintErrors = xlPrintErrorsDisplayed
End With
ActiveSheet.Unprotect
ActiveSheet.PrintOut
ActiveSheet.Protect
Next i
Sheets(Current).Activate
Application.ScreenUpdating = True
End If
End Sub

Weights = 1
' WEIGHTS MODULE
' Most of these subroutines are used by the Weights spreadsheet or the
userform that are used to Swing
Weights
Sub ValidateWeights()
    ' Validates the weights on weight sheet, will not let you leave the sheet
until all the branches and t
he globals sum to 1
Dim BranchesAreOne, GlobalNotOne As Boolean
BranchesNotOne = Not (EqualOne(Range("C3")) And EqualOne(Range("F3")) And
EqualOne(Range("I3")))
d_
EqualOne(Range("D7")) And EqualOne(Range("F7")) And EqualOne(Range("J7")) And
EqualOne(Range("E11")) And EqualOne(Range("H11")) And EqualOne(Range("J11"))
A
nd_
EqualOne(Range("D15")) And EqualOne(Range("H15")) And _
EqualOne(Range("E19")) And EqualOne(Range("G19")))
GlobalNotOne = Not (EqualOne(Range("C1")))
If BranchesNotOne Or GlobalNotOne Then
Msg = "One or more of the Branches weights or the Overall Weight does not sum
to 1.00!" & vbCrLf & "Please review all local weighting."
Ans = MsgBox(Msg, vbOKOnly, "Weights do not sum to 1")
Else: Call ReturnToHierarchy

```

```

End If
End Sub
Function EqualOne(Weight) As Boolean
' Function so that weights do NOT have to EXACTLY equal 1.00. Within 1/1000th
is close enough
If (Weight > 0.998) And (Weight < 1.002) Then
EqualOne = True
Else: EqualOne = False
End If
End Function
Sub ShowSortedGlobal()
' For click button that goes from Value Hierarchy to Show Global Weights Chart
' Ordered Global Weights are stored on the Weights sheet in row 105 and are
used for MANY SUBROUTINES
' Here they are copies to row 109 to be sorted and displayed in the sorted
global weights chart
Application.ScreenUpdating = False
Worksheets("Weights").Activate
ActiveSheet.Unprotect
Range("B105:AB106").Copy Range("B109")
Range("B109:AB110").Select
Selection.Sort Key1:=Range("B110"), Order1:=xlDescending, Header:=xlGuess _
, OrderCustom:=1, MatchCase:=False, Orientation:=xlLeftToRight, _
DataOption1:=xlSortNormal
ActiveSheet.Protect
Charts("Sorted Global Weights Chart").Visible = True
Worksheets("Hierarchy").Visible = False
Charts("Global Weights Chart").Visible = False
Charts("Sorted Global Weights Chart").Activate
Application.ScreenUpdating = True
End Sub

```

Bibliography

- Air Force Studies and Analysis Agency, Future Capabilities Branch, “Capability-Based Logistics Planner”. 1 May 2006.
- Air Mobility Command, “Airfield Suitability Database.” On-line database. n. pag. <https://www.afd.scott.af.mil>. 26 April 2006.
- Clemen, R. and T. Reilly. *Making Hard Decisions with DecisionTools 2004*, Pacific Grove, CA: Duxbury, 2001.
- Google Earth website, <http://earth.google.com>. 26 April 2006.
- Harbaugh, R. *Skill Signaling, Prospect Theory, and Regret Theory*. Claremont Colleges, 2003.
- Hoare, R. “World Climate: Weather, Rainfall, and Temperature Data.” On-line database. n. pag. <http://www.worldclimate.com>. 25 April 2006.
- Joint Publication 3-17. Joint Doctrine and Joint Tactics, Techniques, and Procedures for Air Mobility Operations. Joint Chiefs of Staff. Department of Defense, 2005.
- Keeney, R. *Value Focused Thinking: A Path to Creative Decision Making*, Cambridge, MA: Harvard University Press, 1992.
- Kelton, W.D., R Sadowski, and D. Sturrock. *Simulation with Arena*, New York, NY: McGraw-Hill, 2004.
- Kirkwood, C. *Strategic Decision Making: Multiobjective Decision Analysis with Spreadsheets*, Belmont, CA: Wadsworth Publishing Company, 1997.
- McVicker, P. “En Route Strategic Plan” Report to USTRANSCOM/J5, Scott AFB IL: AMC/XP, February 2002.
- Rieger, M. O., and M. Wang. *Cumulative Prospect Theory and the St. Petersburg Paradox*. SonderForschungsbereich. Universitat Mannheim, Germany, 2004.
- Sere, M. *Strategic Airlift En Route Analysis and Considerations to Support the Global War on Terrorism*, MS Thesis, AFIT/GOR/ENS/05-17. School of Engineering and Management, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, March 2005.
- Tharaldson, M. *Strategic Airlift En Route Analysis to Support the Global War on Terrorism Using a Value Focused Thinking Approach*, MS Thesis, AFIT/GOR/ENS/06-19. School of Engineering and Management, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 2006

Tversky, A. and D. Kahneman. Advances in Prospect Theory: Cumulative Representation of Uncertainty, *Journal of Risk and Uncertainty*, Vol 5, Issue 4, 297-323, 1992.

United States Air Force website, <http://www.af.mil/main/welcome.asp>. 3 April 2006.

United States Air Force, *Air Mobility Planning Factors*, AFPAM 10-1403, Washington: HQ USAF, 18 December 2003.

U.S. Department of State website, <http://www.state.gov/r/pa/ei/bgn>. 26 April 2006.

REPORT DOCUMENTATION PAGE

*Form Approved
OMB No. 074-0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY) 11-05-2006				2. REPORT TYPE Master's Graduate Research Project		3. DATES COVERED (From – To) Jan 2006 – May 2006	
4. TITLE AND SUBTITLE GLOBAL EN ROUTE BASING INFRASTRUCTURE LOCATION MODEL				5a. CONTRACT NUMBER			
				5b. GRANT NUMBER			
				5c. PROGRAM ELEMENT NUMBER			
6. AUTHOR(S) Miravite, Alexander, Jr., Major, USAF Schlegel, Charles F., Major, USAF				5d. PROJECT NUMBER			
				5e. TASK NUMBER			
				5f. WORK UNIT NUMBER			
7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Street, Building 642 WPAFB OH 45433-7765				8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/IOA/ENS/06-08			
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Lt Col Diane Allen, US TRANSCOM JT/TCJ5-AS 508 Scott Drive Scott Air Force Base, IL 62225-5357 e-mail: diane.allen@ustranscom.mil commercial: (618)-229-1459				10. SPONSOR/MONITOR'S ACRONYM(S)			
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)			
12. DISTRIBUTION/AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.							
13. SUPPLEMENTARY NOTES							
14. ABSTRACT The backbone of the air transportation system for the US military is the En Route System (ERS), a collection of personnel and equipment at bases throughout the world. The location of these bases in the ERS bears the marks of the Cold War legacy of the latter half of the twentieth century. While many changes to the ERS have taken place in the last 20 years, it still has a predominantly East-West orientation in the Northern Hemisphere with little capability in many parts of the world. This research provides the decision makers (DMs) with a clear methodology for evaluating the best locations for new en route bases in the ERS. It looks at the capability of each en route airfield alternative using 27 different measures. It then examines the National Security Import of the destination regions an en route airfield could service, along with the probability of events in those regions requiring military action, including humanitarian and other missions. This is accomplished using Value Focused Thinking (VFT) and decision analysis tools, with the goal of providing insight to the DMs to make informed decisions. This methodology is then packaged in a powerful Excel-based, user interface, allowing quick analysis of multiple scenarios. GERBIL is currently being used by TRANSCOM/J5 to prioritize Cooperative Security Locations around the globe for possible TRANSCOM mobility en route locations.							
15. SUBJECT TERMS Value Focused Thinking; En Route System; Mobility; Basing							
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 213	19a. NAME OF RESPONSIBLE PERSON Raymond W. Staats, Lt Col, USAF (ENS)		
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U			19b. TELEPHONE NUMBER (<i>Include area code</i>) (937) 255-6565, ext 4518 e-mail: raymond.staats@afit.edu		

Standard Form 298 (Rev. 8-98)

Prescribed by ANSI Std. Z39-18